Supplementary: United We Stand, Divided We Fall: UnityGraph for Unsupervised Procedure Learning from Videos

Siddhant Bansal*
CVIT, IIIT, Hyderabad

Chetan Arora
IIT, Delhi

C.V. Jawahar
CVIT, IIIT, Hyderabad

1. Appendix

This document contains additional results and analysis to support the main paper. In Section 2, we first compare the results obtained using the proposed GPL framework with CnC for fewer videos. Then, in Table 2, we show the effectiveness of using background frame removal using hand-object interaction. Finally, in Section 3, we visualise and analyse UnityGraph's embeddings obtained using t-SNE [5].

2. Additional Results

Effect of number of videos to train a method: In Table 1, we compare the representation learned by GPL from CnC [1] by increasing the number of videos for the same task. The table contains results on the Bacon and Eggs category from EgoProceL. As can be seen in the table, the results of both GPL and CnC increase upon increasing the number of videos. However, due to the novel UnityGraph– that creates both spatial and temporal connections – the results for the less number of videos using the proposed GPL framework are high. This shows the effectiveness of modeling multiple videos of the same task as a single graph. On the contrary, CnC [1] learns from a pair of videos at a time, leading to learning limited information and hence, low results.

Effect of Background frame removal: In Table 2, the results obtained with and without background frames on EgoProceL are presented. As was concluded in the main paper, the results improve for the categories that involve subjects moving around in an unrestricted space. The reason for this is people working in an unrestricted space need to walk around looking for objects leading to background frames. As there is no hand-object interaction present in the background frames, using Shan *et al.*'s [4] hand-object detector, we can filter them.

We observe significant improvement in the results for EGTEA Gaze Plus and EPIC-Tents, as the tasks being performed there are in an unconstrained environment, kitchen and lawn, respectively. This allows the subjects to search for the objects and wander around during the process. Also, we observe marginal improvement on CMU-MMAC where the subjects are allowed to work in an unconstrained environment (kitchen) but are constrained by the wires used to capture various modalities. Due to this, there are on average few sections where the person is not interacting with an object. On the contrary, no improvements are observed for PC assembly, PC disassembly, and MECCANO. This is because, here, the subjects are working in a constrained environment – table top and CPU – restricting the movement. Hence, very low number of frames where there is no hand-object interaction.

Amount of background frames removed: Table 3 contains the amount of frames removed (in percent) upon using Shan *et al.*'s hand-object detector [4]. It can be seen that the amount of frames without hand-object interaction is proportional to the gain in results (in Table 2).

3. UnityGraph Embedding Space's Visualisation

In this section, we discuss the t-SNE [5] visualisation in Figure 1 generated from UnityGraph's embeddings before and after applying the Node2Vec [3] algorithm. On the left-hand side of Figure 1, t-SNE visualisation for UnityGraph's

Table 1. **Number of Videos**. The results are obtained upon systematically increasing the number of videos for creating UnityGraph. Here, we compare the performance of UnityGraph over CnC [1]. Numbers in **bold** represent the highest number in the column and underlined number represent the highest number in the row for that metric. **R**, and **F** represent recall, and F-score, respectively

Number of Videos	UnityGraph (*ours*)			CnC [1]		
	R	**F**	**IoU**	**R**	**F**	**IoU**
4	23.6	20.0	11.4	18.8	16.3	4.4
8	25.0	22.1	12.1	20.3	17.9	6.1
16	**27.8**	**23.1**	**12.6**	21.2	18.6	9.3

*Corresponding author: siddhant.bansal@research.iiit.ac.in

Table 2. **Detecting the background frames.** Here, the results are obtained upon filtering the frames that do not contain hand-object interaction. For the categories that involve the subject moving in an unrestricted space and performing the tasks, the results improve. For example, CMU-MMAC and EPIC-Tents. This shows that detecting hand-object interaction is an efficient technique for identifying the background frames. Note that the results even without removing the background frames improve over state-of-the-art method [1] for most of the categories. This highlights the efficacy of the proposed GPL framework

	EgoProceL											
	CMU-MMAC		EGTEA G.		MECCANO		EPIC-Tents		PC Assembly		PC Disassembly	
	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU	F1	IoU
CnC [1]	22.7	11.1	21.7	9.5	18.1	7.8	17.2	8.3	25.1	12.8	**27.0**	14.8
Not Checked	30.2	16.7	23.6	14.9	20.6	9.8	18.3	8.5	**27.6**	14.4	26.9	15.0
Checked	**31.7**	**17.9**	**27.1**	**16.0**	**20.7**	**10.0**	**19.8**	**9.1**	27.5	**15.2**	26.7	**15.2**

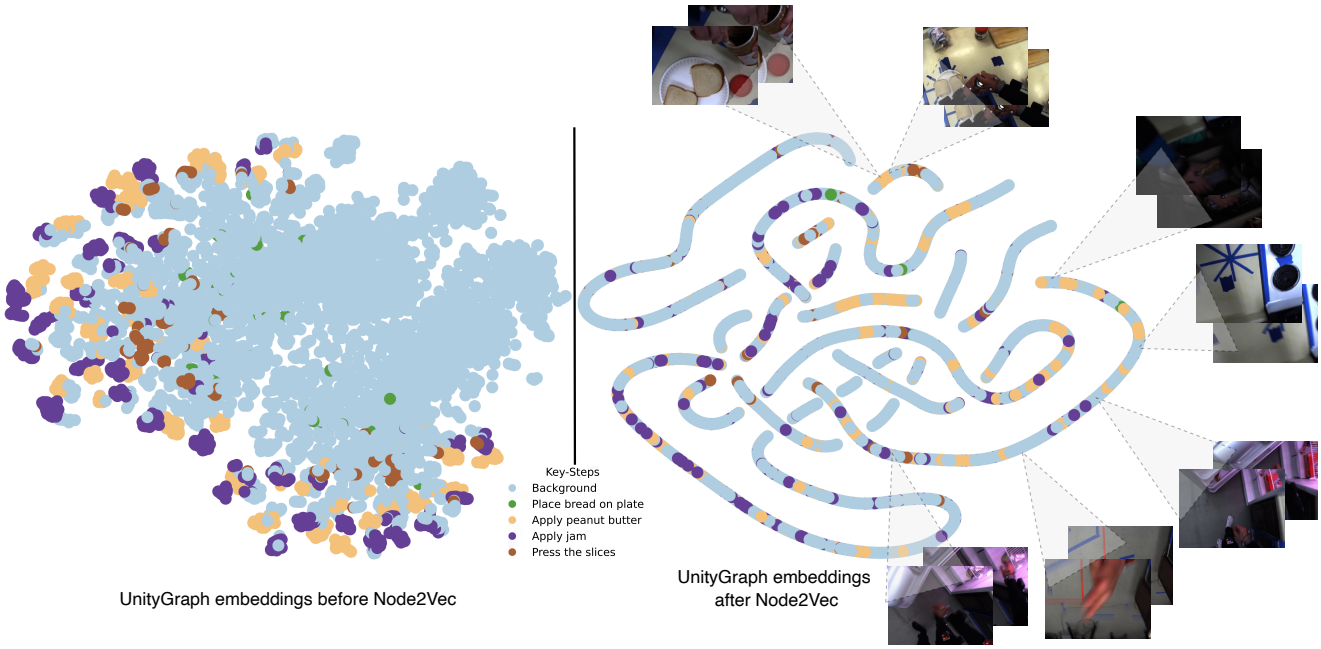

Figure 1. **t-SNE [5] visualisation** for the task of making a sandwich [1, 2] before and after updating UnityGraph's embeddings using the Node2Vec algorithm [3]. Here, each color represents a key-step's category as noted in the legend. The left side of the figure consists of t-SNE visualisation obtained before using the Node2Vec algorithm. The right side of the figure consists of t-SNE visualisation obtained after updating UnityGraph's embeddings using Node2Vec. As can be seen, upon updating the embeddings using Node2Vec, clips with similar key-steps come close. For example, the cluster on the top consists of clips of subjects applying peanut butter, whereas the cluster towards the centre has background clips of subject moving themselves from one place to other.

clip-level embeddings before applying the Node2Vec algorithm are shown. Though UnityGraph's embeddings are able to capture the background clips well, they fall short of bringing the key-steps close in the embedding space. As can be seen, the "apply jam" key-step is spread across the embedding space.

On the other hand, on the right-hand side of Figure 1, t-SNE visualisation for UnityGraph's clip-level embeddings after applying the Node2Vec algorithm is shown. Here, we can see that the embeddings have arranged themselves in a particular pattern. In the majority of the cases, the embeddings for similar key-steps have come closer. For example, the cluster on the top consists of subjects applying peanut butter. The clips for "apply jam" have concentrated in a few selected clusters. And due to the nature of UnityGraph (connecting clips with similar semantic information), background clips with similar styles of information have come closer. For example, the cluster towards the center has background clips of subjects moving from one location to another.

Table 3. Background frame removal percentage. Here we show the amount of frames removed from various EgoProceL's tasks after using Shan *et al*.'s [4] hand-object detector

Category	% background frames
CMU-MMAC	18
EGTEA G.	29
MECCANO	4
EPIC-Tents	12
PC Assembly	2
PC Disassembly	2

References

[1] Siddhant Bansal, Chetan Arora, and C.V. Jawahar. My View is the Best View: Procedure Learning from Egocentric Videos. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 2

[2] F. De La Torre, J. Hodgins, A. Bargteil, X. Martin, J. Macey, A. Collado, and P. Beltran. Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) database. In *Robotics Institute*, 2008. 2

[3] Aditya Grover and Jure Leskovec. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. 1, 2

[4] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3

[5] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 2008. 1, 2