Supplementary
Assembly101: A Large-Scale Multi-View Video Dataset for Understanding Procedural Activities

Fadime Sener[†] Dibyadip Chatterjee[‡] Daniel Shelepov[†] Kun He[†] Dipika Singhania[‡]
Robert Wang[†] Angela Yao[‡]

[†] Meta Reality Labs Research [‡] National University of Singapore
{famesener,dsh,kunhe,rywang}@fb.com {dibyadip,dipika16,ayao}@comp.nus.edu.sg
https://assembly101.github.io/

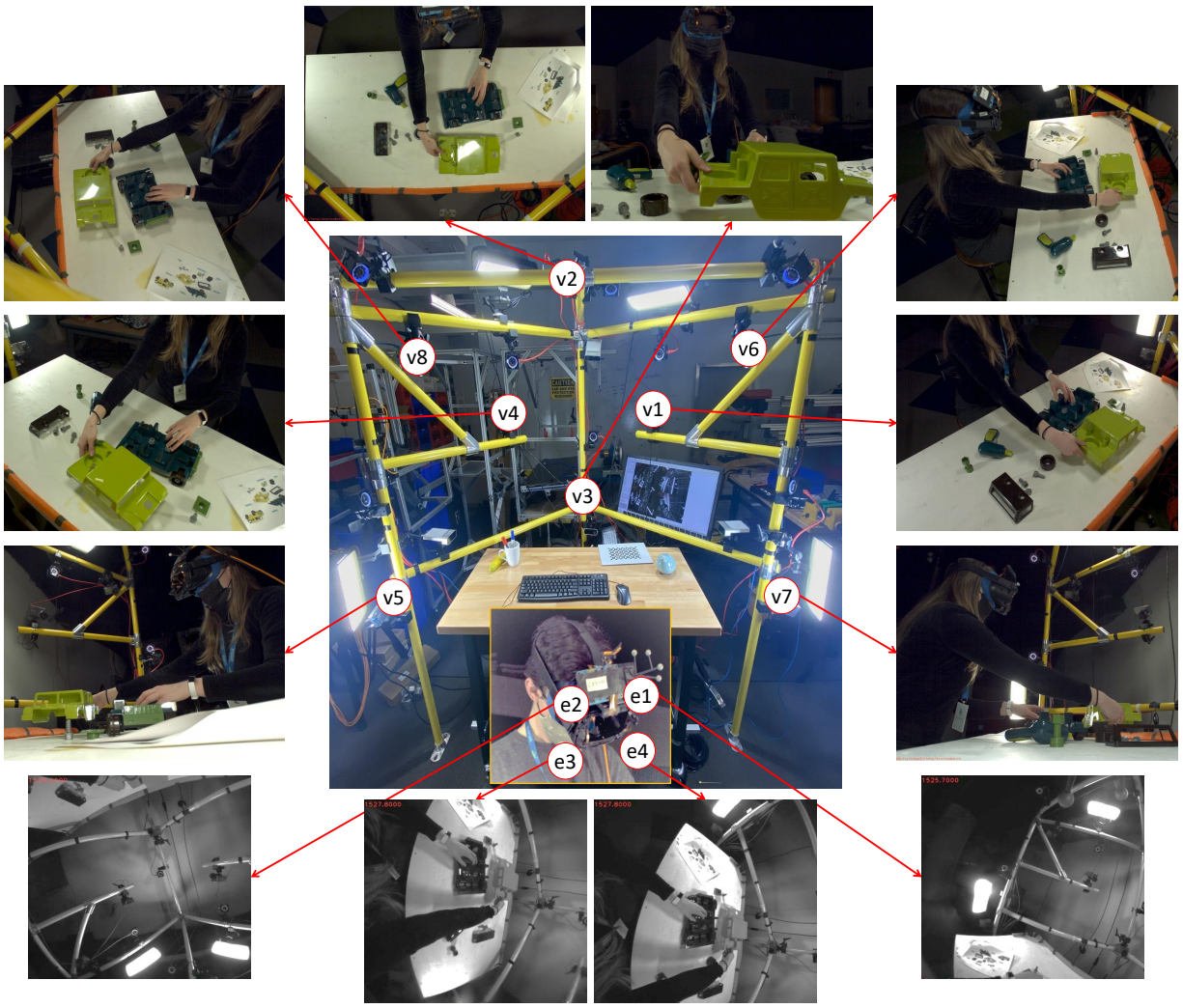

Figure 1. Our desk-based rig and sample frames from eight RGB and four monochrome cameras.

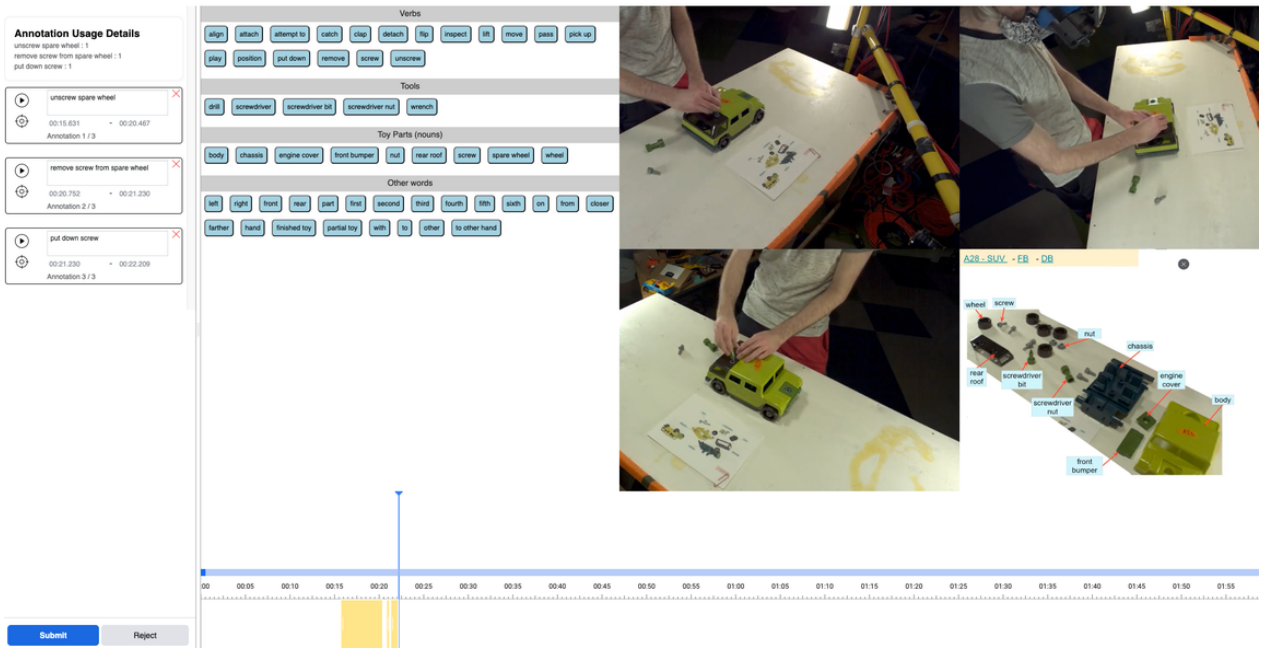

Figure 2. Our annotation tool interface. **Right panel**: input video composed of three static camera views and a diagram of the objects. **Middle panel**: pre-defined lists of verbs, tools and objects for labelling segments. The annotators also have the flexibility for free-form entry. **Bottom bar**: this annotation bar shows the temporal boundaries of the actions, i.e., the start and end of each action. **Left panel**: list of temporally annotated actions.

This supplementary further details recording settings, annotations and experiments. Section 1 provides an overview of annotator training, our recordings and custom interface. Section 2 provides the distributions of our labels, detailed train/validation/test statistics, and an extensive set of comparisons with other related datasets. Section 3 presents the architecture and implementation details of our baselines. Finally, in Section 4, we present more results and comparisons on our baselines.

1. Recording and Annotation

1.1. Recording rig

We built a dedicated desk-based rig to capture the sequences in this dataset. Each sequence is recorded with eight RGB cameras at 1920 × 1080 resolution and four monochrome cameras at 640 × 480 resolution. Fig. 1 shows individual camera views and sample frames.

1.2. Participants

We recruited 53 adult participants (28 males, 25 females) to record approximately one-hour sessions over the course of 18 consecutive days. Participants were recruited considering the guidelines and restrictions of COVID-19, including wearing face masks. We obtained informed consent of camera wearers for the digital capture of participants, which means digitally capturing participants' faces and bodies. All video footage and collected annotations are available for the research community.

1.3. Annotations

Annotation interface: We developed a custom interface (see Fig. 2) for annotators to temporally locate the start- and end-frames of fine-grained action segments. Each action segment is tagged with predefined verbs, tools and objects, though annotators also have the flexibility for free-form entry. To promote precise annotations, we displayed three static camera views to ensure that the actions are visible at least from one view without self-occlusion from the working hand. Additionally, we provided diagrams for the annotators with labelled objects of all 101 toys to ensure correct naming and terminology. While working with the toys, the participants were requested to simultaneously describe out loud their actions with named tools and objects, *e.g. "I am flipping the truck over and putting the right front wheel on the truck"*. To assist with the description, the completed toy in the reference diagrams are labelled with part names.

Annotator training: To ensure high-quality labels, we trained annotators over the course of four days. During this time, the annotators were introduced to our interface and the labelling task under the authors' guidance. After training,

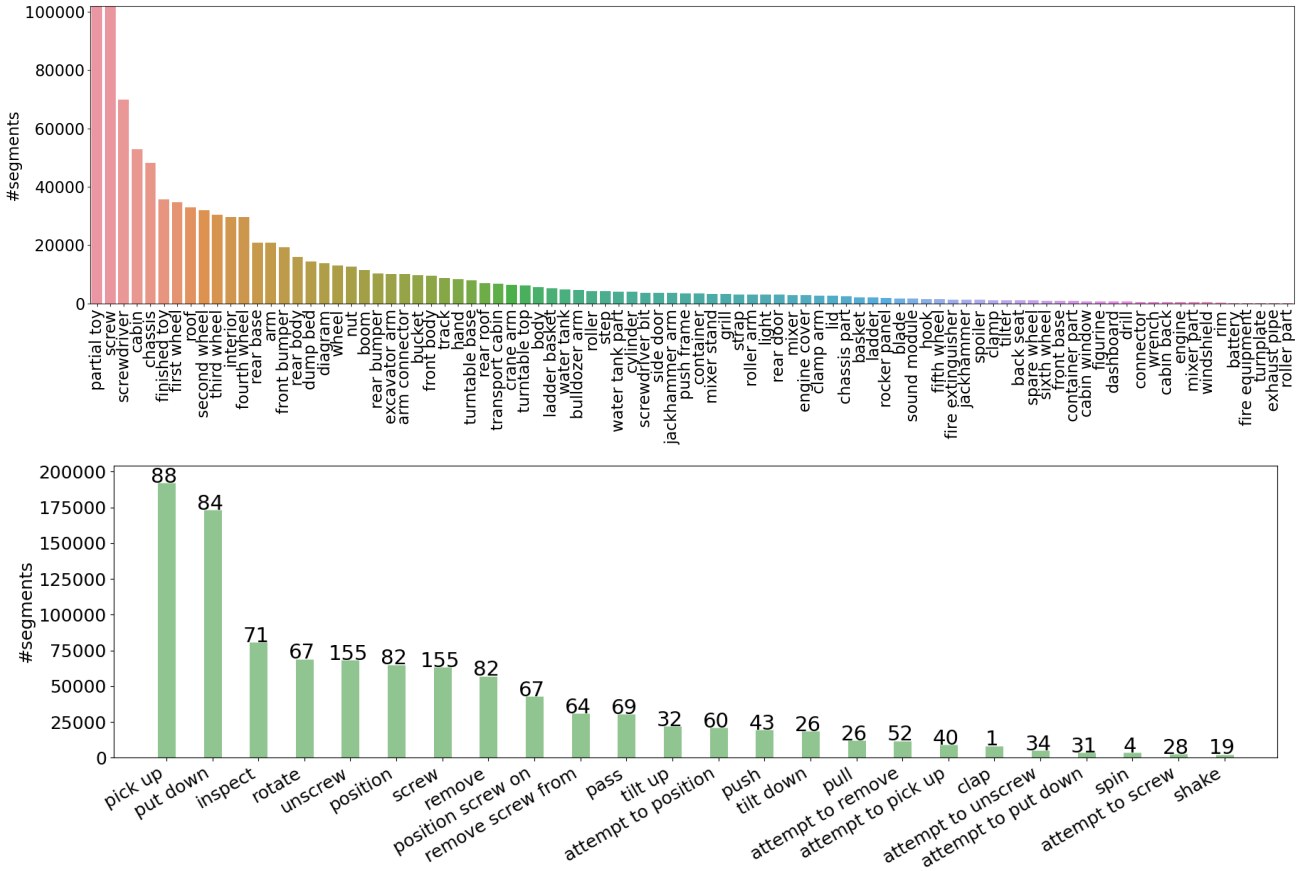

Figure 3. We define 90 objects (**upper**) and specify 24 verbs (**bottom**), forming a total of 1380 fine-grained action labels. The verb distribution also shows the number of actions containing that verb on top of each bar.

annotators who were slow or made many mistakes were not selected to continue. Following the training, the labelling was completed by 21 annotators over 213 hours of work.

2. Dataset Statistics & Splits

2.1. Fine-grained actions

From our 15 toy categories, we define 90 unique objects. Additionally, we define 24 verbs. Six of the 24 verbs are used to describe "attempts", *i.e.* the participants adjust or change their minds during assembly. For example, the "pick up chassis" action is composed of three stages of *reaching for the chassis*, *grasping it*, and *lifting it up*. Our annotators were provided with the stages of each verb. When users do not complete all stages in a segment, *e.g.* approach and/or grasp the chassis but do not lift it, we asked our annotators to place "attempt to" in front of the action. The objects and verbs combined form a total of 1380 fine-grained action labels as not every possible combination is observed. We present the distribution of our verbs and objects in Fig. 3.

To highlight the scale of our dataset, we compare Assembly101 to other video datasets for action recognition in Table 1. Our dataset is the largest in number of segments and the richest in terms of multi-view recordings both from third-person and egocentric views.

Balancing the head: The objects, verbs, and fine-grained action labels each naturally form a long-tailed distribution [33]. When reviewing Meccano and IKEA, we observe that a handful of head-classes dominate the action distribution (60% of actions belong to 3 head classes in IKEA, and 30% in Meccano). To mitigate similar effects, we made two labelling design choices concerning the wheels, screws and tools, as they are the most commonly occurring object parts. The adjustments spread the head-tail distribution (the top 3 classes account for only 13% of the action segments) and add semantic richness to the dataset:

- **Enumerating the wheels:**, *i.e. "position first wheel"* vs. the generic *"position wheel"* action. Enumeration also extends the range of temporal dependencies in a sequence, as algorithms must keep track of how many wheels have been attached or removed.

- **Fine-grained tool and screw verbs:** Due to the nature of the assembly task, tools and screws appear very frequently. To spread the head classes that result from treating tools and screws as simple objects or parts, we introduced dedicated verbs, *e.g.* "screw [object] with drill", "position screw on [object]" and "remove screw from [object]". Coupling these verbs with other objects conveys more information than "screw chassis" or "position screw".

2.2. Coarse actions

Each coarse action is defined by the assembly or disassembly of a vehicle part. There are 202 coarse actions composed of 11 verbs and 61 objects. Each video sequence features an average of 24 coarse actions. There is an average of 10 fine-grained actions per coarse action segment. The average number of coarse actions is 14 in each assembly sequence and 10 in each disassembly sequence. Table 2 compares Assembly101 with other video datasets with coarse labels. Our dataset is the largest in video hours and number of segments, and the only non-cooking recorded dataset.

2.3. 3D hand poses

Action recognition from 3D hand poses is much less explored compared to the full human body. The only existing datasets [10, 16] that focus on hand-object action recognition with 3D hand pose annotations are small-scale and/or include only a single hand [10]. We present our comparisons in Table 3. Compared with FPHA [10] and H2O [16], our dataset includes 82× more action segments and 200× more frames. We also compare the scale of our dataset with NTU RGB+D 60 [29] and NTU RGB+D 120 [20], which are the largest full-body pose dataset. Our dataset contains 6-12× more action classes and 27-13× more frames. Additionally, NTU RGB+D 60 and NTU RGB+D 120 are composed of short trimmed clips of actions while our segments are related to each other with sequence dynamics, which allows for studying the importance of temporal context for action recognition.

2.4. Training, validation & test splits

We use a 60/15/25 split of recordings for dividing our dataset into training, validation and test splits, with detailed statistics presented in Table 5. We present the distribution of the mistake action in Table 4.

For evaluation purposes, we will hold out the ground truth annotations of the test split. These will be used for online challenge leaderboards to track future progress on our target tasks. Our dataset is designed to assess the generalizability to new toys, actions and the participants' skills. We thus structured our validation and test sets to examine models under varying conditions.

Seen/Unseen vehicles/toys: Of the 101 toys, only 25 toys are shared across all the three splits. We designed the splits to ensure that there are unseen toys in the training to facilitate zero-shot learning. There are 20 and 16 unseen toy instances in the validation and test splits, respectively.

Head vs. tail classes: The distribution of our objects and verbs can be seen in Figure 3. There is a large number of common manipulation verbs such as "pick up" and "put down", which naturally depicts a long tail distribution. The object and action distribution follow the same general trend. We define the tail classes as the set of action classes whose instances account for 30% of the training data. This amounts to 1238 (89%) tail action classes. We used Epic-Kitchens as a reference when forming our tail classes, where 87% of the action classes are in the tail. Similarly, we define the tail classes of the coarse labels as the set of coarse action classes whose instances account for 30% of the training data. This amounts 171 (84%) tail action classes.

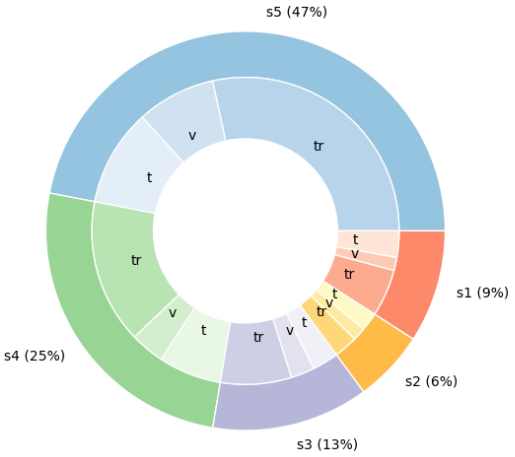

Figure 4. The distribution of skill level of the participants from 1 (the worst) to 5 (the best). Overall, 9% of the sequences are from the participants with the worst skill level and 47% is from the best. 'tr', 'v' and 't' stand for the training, validation and test splits.

Skill level assessment is a critical task in many areas including sports [22], robot learning [35], surgery [38] and assembly line [24]. Which participant has the highest assembly skills? How are the participants progressing with more assembly tasks? What are the common mistakes made by participants? Answering these questions involves determining how well the assembly was carried out. Thus, we annotated the skill levels of the participant in each video from 1 (the worst) to 5 (the best). Skill level criteria is based on the participant's assembly speed and number of mistakes, with coarse thresholds. Overall, the distribution of skill labels in our sequences is 9%, 6%, 13%, 25% and 47% from the worst to the best (see Fig. 4).

Table 1. Comparison with other video datasets for action recognition on fine-grained actions.

Dataset	total hours	# videos	# segments	# actions	recorded	multi-view	egocentric	#pose annotation	year
MPII [27]	8.3	44	5,609	64	✓	✗	✗	✓	2012
ActivityNet [3]	648.0	27,811	23,064	200	✗	✗	✗	✗	2015
Charades [32]	81.1	9,848	67,000	157	✓	✗	✗	✗	2016
THUMOS [13]	30.0	5,613	6,310	101	✗	✗	✗	✗	2017
Charades-EGO [31]	68.8	2,751	30,516	157	✓	✓	✓	✗	2018
EPIC-100 [5]	100.0	700	89,977	4053	✓	✗	✓	✗	2020
H2O [16]	5.5	186	934	11	✓	✓	✓	✓	2021
Meccano [26]	6.9	20	8,858	61	✓	✗	✓	✗	2021
IKEAASM [2]	35.0	371	17,577	33	✓	✓	✗	✓	2021
Ego4D [11]	120.0	-	77,002	-	✓	✗	✓	✗	2021
Assembly101	513.0	4,321	1,013,523	1380	✓	✓	✓	✓	2021

Table 2. Coarse action label dataset comparisons.

Dataset	hours	#videos	#segments	#actions	#recorded	#multi-view	#egocentric	#cooking	#year
GTEA [7]	0.4	28	500	71	✓	✗	✓	✓	2011
50Salads [36]	4.5	50	899	17	✓	✗	✗	✓	2013
Breakfast [15]	77.0	1,712	11,300	48	✓	✓	✗	✓	2014
YouTube Instructional [1]	7.0	150	1,260	47	✗	✗	✗	✗	2016
COIN [37]	476.0	11,800	46,000	778	✗	✗	✗	✗	2019
CrossTask [41]	374.0	4,700	34,000	107	✗	✗	✗	✓	2019
YouCookII [40]	176.0	2,000	15,400	-	✗	✗	✗	✓	2018
Assembly101	513.0	4,321	104,759	202	✓	✓	✓	✗	2021

Table 3. Comparisons with other datasets with 3D hand pose.

Dataset	Hours	#frames	#segments	#actions
NTU RGB+D 60 [29]	-	4.0M	56K	60
NTU RGB+D 120 [20]	-	8.0M	114K	120
FPHA [10]	1.0h	0.1M	1K	45
H2O [16]	5.5h	0.5M	1K	36
Assembly101	513.0h	110.0M	86K	1380

Table 4. The distribution of {*"correct", "mistake", "correction"*} segments on the coarse segments of the assembly sequences.

	#correct	#mistake	#correction
Test	12,337	3,144	1,268
Validation	8,984	1,624	640
Train	25,718	4,941	2,226
Overall	47,039	9,709	4,134

3. Implementation Details

We define four action challenges: recognition, anticipation, temporal segmentation, and mistake recognition.

3.1. Action recognition

3.1.1 Appearance-based action recognition

Top-performing video-based action recognition models [4, 8] are typically extensions of state-of-the-art image-based architectures [12]. Some works extend convolution and pooling to the time dimension [4, 8]; others perform channel shifting [6, 18] to capture temporal relationships while maintaining the complexity of a 2D CNN. We adopted a state-of-the-art model, TSM [18], as the baseline for this task.

Implementation details: We use two versions of the standard TSM architecture with a ResNet-50 [12] backbone — one with a single classifier head for predicting the actions and another with two classifier heads for predicting the objects and verbs separately. Both models are trained using stochastic gradient descent (SGD) with a momentum of 0.9, weight decay of 0.0005, and dropout of 0.5 for 50 epochs with a batch size of 64. The learning rate is initialized as 0.001 and decayed by a factor of 10 at epochs 20 and 40. The best-performing model is selected via early-stopping over the validation set. Sampling and augmentation during training and inference for TSM is done following [5].

Table 5. Statistics of Assembly101 and its Train/Validation/Test splits.

Split	Hours	#videos	#unseen toys	#shared toys	#fine segments	#fine verbs	#fine objects	#fine actions	#coarse segments	#coarse verbs	#coarse objects	#coarse actions
Train	287.6	2526	40	26	566,855	24	85	1244	57,657	11	59	195
Validation	96.6	740	16	18	186,788	24	81	1018	19,008	10	56	164
Test	128.8	1055	20	20	259,880	24	79	1045	28,094	11	55	172
Overall	513.0	4321	76	25	1,013,523	24	90	1380	104,759	11	61	202

3.1.2 Pose-based action recognition:

State-of-the-art methods for recognizing skeleton-based actions are based on deep architectures such as CNNs [19], transformers [25] and graph convolutional networks (GCN) [21, 39]. We use two state-of-the-art GCN-based methods for our experiment, 2s-AGCN [30] and MS-G3D [21].

Implementation details: We use the publicly available PyTorch [23] code for 2s-AGCN and MS-G3D. All hand pose sequences are padded to T = 200 frames by replaying the action segments. If there is one hand missing, we pad the second hand with 0. No data augmentation is used.

We trained 2s-AGCN [30] using SGD with Nesterov momentum of 0.9 and a learning rate of 0.1 with a batch size of 32 for 30 epochs. The weight decay is set to 0.0001. For MS-G3D [21], we used SGD with a momentum of 0.9 and a learning rate of 0.05. We set the batch size to 16 and the weight decay to 0.0005. The model is trained for 50 epochs.

3.2. Action anticipation

In our experiments, the anticipation task is defined as predicting the upcoming fine-level actions *1 second* before they start. We adopted TempAgg [28] as baseline for this task. Similar to previous works [5, 9], we report class-mean Top-5 recall as it accounts for uncertainty in future predictions.

Implementation details: We use the TempAgg with three classification heads that predicts objects, verbs and actions separately. Since TempAgg operates on frame features, we use the 2-D backbone of the TSM fine-tuned on our dataset to extract the 2048-D frame features. The *spanning past* snippet features are computed over a period of 6 seconds before the start of the action and aggregated at 3 temporal scales $K = \{5, 3, 2\}$. The *recent past* snippet features are computed over a period of $\{1.6, 1.2, 0.8, 0.4\}$ before the start of the action and aggregated over a single temporal scale $K_R = 2$. The model is trained using an Adam [14] optimizer for 15 epochs with a batch size of 32. A dropout factor of 0.3 is used. The learning rate initialised as 0.0001 and decayed by a factor of 10 after the 10^{th} epoch.

3.3. Temporal action segmentation

For temporal action segmentation, we apply two competing state-of-the-art temporal convolutional networks: MS-TCN++ [17], which maintains a fixed temporal resolution in its feed-forward structure with successively larger kernel dilation, and C2F-TCN [34], a U-net-style shrink-then-expand encoder-decoder architecture. For C2F-TCN, we use implicit ensembling of decoder layers and the feature augmentation strategy detailed in the paper. Performance is evaluated by mean frame-wise accuracy (MoF). Since longer actions dominate this score and it does not penalize over-segmentation errors explicitly, we also report segment-wise edit distance (Edit) and F1 scores at overlapping thresholds of 10%, 25%, and 50%, denoted as by F1@10, 25, 50.

Implementation details: For both C2F-TCN [34] and MS-TCN++ [17], we use an Adam [14] optimizer with a batch size of 20 for a maximum of 200 epochs while using early-stopping to select the model that best fits the validation data. Loss functions used for both models are frame-wise cross entropy loss weighted with 1 and mean-square error loss [17] weighted with 0.17. For MS-TCN++, we use a learning of 0.0005 and a weight decay of 0. For C2F-TCN, we use a learning rate of 0.001 and weight decay of 0.0001. The base window for feature augmentation sampling is set to be 20 and all layers of decoder are included in ensembling.

3.4. Mistake detection

We introduce the new problem of mistake detection in assembly videos. We adopted TempAgg [28] as the baseline for this task, which captures long-range relationships that span an order of several minutes successfully.

Implementation details: We modified the TempAgg model to capture even longer-range relationships. More precisely, the *spanning past* snippet features are computed over a period of 60 seconds around the action segment, i.e., $[s - 60, e + 60]$, aggregated at 3 temporal scales $K = \{5, 3, 2\}$, where s and e are the start and end timestamps of the action in seconds. The *recent past* snippet features are computed over a period of $\{3.0, 2.0, 1.0, 0.0\}$ around the action segment and aggregated over a single temporal

Table 6. **Action recognition** on fine-grained actions evaluated by Top-5 accuracy.

		Overall			Head			Tail			Seen Toys			Unseen Toys		
Task	Tested on	verb	object	action	verb	object	action	verb	object	action	verb	object	action	verb	object	action
Recognition	Fixed	91.2	77.0	63.3	93.8	89.6	78.0	84.9	45.8	26.4	90.8	84.8	68.6	91.4	74.6	61.6
	Egocentric	82.7	64.3	44.3	86.0	79.1	57.8	74.6	27.4	10.8	83.5	67.8	46.3	82.5	63.2	43.7
	Fixed & Ego.	88.5	72.9	57.1	91.2	86.2	71.4	81.6	39.8	21.3	88.4	79.2	61.2	88.5	70.9	55.8

Table 7. **Action recognition & anticipation** performance on fine-level actions (evaluate by Top-1 acc. and Top-5 recall respectively) using TSM and TempAgg respectively. "Fusion" corresponds to average-pooling the scores from multiple views.

		Overall			Head			Tail			Seen Toys			Unseen Toys		
		verb	object	act.	verb	object	act.	verb	object	act.	verb	object	act.	verb	object	act.
Recognition	Overall	58.5	45.2	34.0	63.7	57.2	44.6	45.3	15.1	7.3	57.8	48.9	35.9	58.7	44.0	33.3
	Fusion	71.6	59.0	48.0	77.4	74.4	63.2	57.0	20.9	10.4	71.0	64.7	51.2	71.8	57.2	46.9
Anticipation	Overall	55.1	29.4	8.8	58.5	55.3	28.0	51.6	29.1	5.3	54.3	43.5	13.9	55.3	22.8	7.3
	Fusion	59.2	31.3	9.1	62.6	62.3	34.8	55.5	30.3	4.5	58.3	48.3	15.7	59.4	23.4	7.8

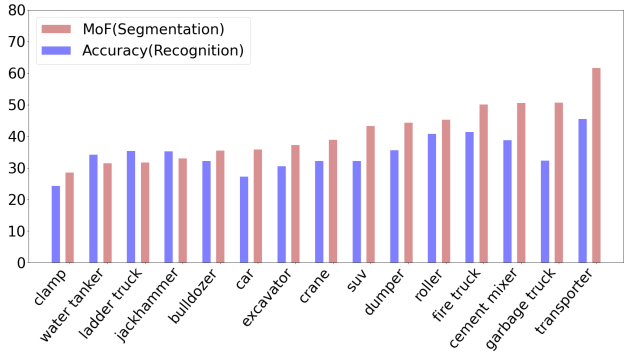

Figure 5. Action recognition accuracy and segmentation MoF over toy categories.

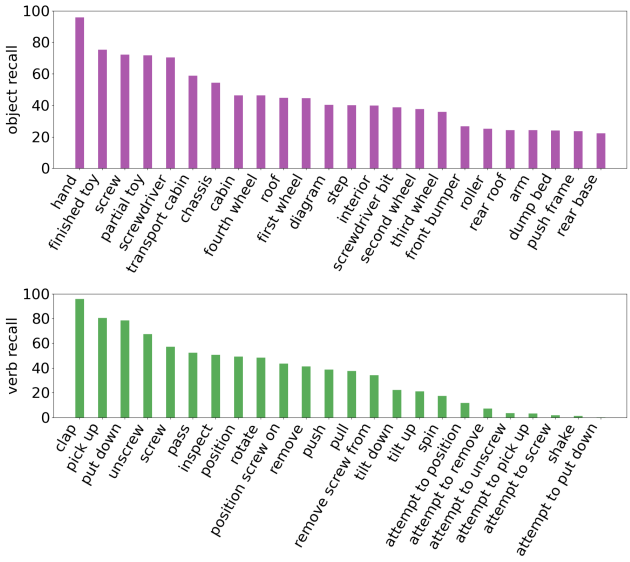

Figure 6. Action recognition object and verb recall.

scale $K_R = 5$. The training scheme remains similar to anticipation, i.e., it is trained on 2048-D TSM features using an Adam [14] optimizer for 15 epochs with a batch size of 32 and a dropout of 0.3 on a single GPU. The learning rate initialised as 0.0001 decayed by a factor of 10 after the 10^{th} epoch. Due to the imbalanced class distribution, we used a weighted cross-entropy loss to penalize the model more for misclassifying "mistake" and "correction" classes.

4. Results

4.1. Action recognition & anticipation

In Table 6, we provide Top-5 accuracy for action recognition. We compare our *"Overall"* performance with results obtained by fusing scores from multiple views on recognition and anticipation in Table 7. The fusion increases the performance of recognition significantly, while the improvement is smaller for anticipation.

4.2. Skill level

We did not observe a significant difference across skill levels for action recognition and anticipation tasks. A reason could be that those tasks are trained on fine-level labels while skill is more relevant for coarse actions.

4.3. Toy categories

Figure 5 shows the accuracy of action recognition and temporal action segmentation models for each toy category. The toy with the highest score is "transporter". Although we have only 4 toys in "transporter" category, there are 22 participants recording these toys. We think its high perfor-

mance could be due to the large number of recordings.

4.4. Class-based evaluations

Fine-grained actions. We present the recall of the objects and verbs for action recognition in Fig. 6. The verbs with the highest recall are "clap", "pick up" and "put down", while the tail verbs involving "attempt to" have the lowest recall. We also present the top 24 object classes in Fig. 6. It can be seen that enumerated wheels are among the top classes.

Coarse actions. Based on the temporal action segmentation results, we further investigated the performance of verbs and objects. Out of 11 coarse verbs, the verbs with the highest recall are "demonstrate", "attach" and "detach", and the ones with the lowest recall are "position", "remove" and "attempt to screw", which are the tail verbs. The objects with the highest recall are "chassis" and "interior", which are the most common objects across toys.

References

[1] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4575–4583, 2016. 5

[2] Yizhak Ben-Shabat, Xin Yu, Fatemeh Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 847–859, 2021. 5

[3] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015. 5

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6299–6308, 2017. 5

[5] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *CoRR*, abs/2006.13256, 2020. 5, 6

[6] Linxi Fan*, Shyamal Buch*, Guanzhi Wang, Ryan Cao, Yuke Zhu, Juan Carlos Niebles, and Li Fei-Fei. Rubiksnet: Learnable 3d-shift for efficient video action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 5

[7] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 5

[8] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6202–6211, 2019. 5

[9] Antonino Furnari, Sebastiano Battiato, and Giovanni Maria Farinella. Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 6

[10] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 409–419, 2018. 4, 5

[11] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, ayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonz alez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, ablo Arbel aez, David Crandall6, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Around the World in 3,000 Hours of Egocentric Video. *CoRR*, abs/2110.07058, 2021. 5

[12] 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. 5

[13] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017. 5

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6, 7

[15] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 5

[16] Taein Kwon, Bugra Tekin, Jan Stuhmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. *arXiv preprint arXiv:2104.11181*, 2021. 4, 5

[17] Shi-Jie Li, Yazan AbuFarha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. 6

[18] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7083–7093, 2019. 5

[19] Jian Liu, Naveed Akhtar, and Ajmal Mian. Skepxels: Spatio-temporal image representation of human skeleton joints for action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019. 6

[20] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2684–2701, 2019. 4, 5

[21] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 143–152, 2020. 6

[22] Paritosh Parmar and Brendan Tran Morris. Learning to score olympic events. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–28, 2017. 4

[23] 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*, 32:8026–8037, 2019. 6

[24] Mikkel Rath Pedersen, Lazaros Nalpantidis, Rasmus Skovgaard Andersen, Casper Schou, Simon Bøgh, Volker Krüger, and Ole Madsen. Robot skills for manufacturing: From concept to industrial deployment. *Robotics and Computer-Integrated Manufacturing*, 37:282–291, 2016. 4

[25] Chiara Plizzari, Marco Cannici, and Matteo Matteucci. Spatial temporal transformer network for skeleton-based action recognition. In *International Conference on Pattern Recognition*, pages 694–701. Springer, 2021. 6

[26] Francesco Ragusa, Antonino Furnari, Salvatore Livatino, and Giovanni Maria Farinella. The meccano dataset: Understanding human-object interactions from egocentric videos in an industrial-like domain. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1569–1578, 2021. 5

[27] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2012. 5

[28] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 154–171. Springer, 2020. 6

[29] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019, 2016. 4, 5

[30] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12026–12035, 2019. 6

[31] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and observer: Joint modeling of first and third-person videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7396–7404, 2018. 5

[32] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 5

[33] Eero P Simoncelli and Bruno A Olshausen. Natural image statistics and neural representation. *Annual review of neuroscience*, 24(1):1193–1216, 2001. 3

[34] Dipika Singhania, Rahul Rahaman, and Angela Yao. Coarse to fine multi-resolution temporal convolutional network. *arXiv preprint arXiv:2105.10859*, 2021. 6

[35] Bradly C Stadie, Pieter Abbeel, and Ilya Sutskever. Third-person imitation learning. *arXiv preprint arXiv:1703.01703*, 2017. 4

[36] Stein and McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *UbiComp*, 2013. 5

[37] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[38] S Swaroop Vedula, Anand Malpani, Narges Ahmidi, Sanjeev Khudanpur, Gregory Hager, and Chi Chiung Grace Chen. Task-level vs. segment-level quantitative metrics for surgical skill assessment. *Journal of surgical education*, 73(3):482–489, 2016. 4

[39] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018. 6

[40] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018. 5

[41] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5