Understanding Multi-Task Activities from Single-Task Videos

Supplementary Material

1. MEKA Dataset

1.1. Data Collection

To create the **Multi-task Egocentric Kitchen Activities (MEKA)** dataset, we recorded videos of eight participants performing various kitchen activities while wearing HoloLens 2 AR glasses, which provide a first-person perspective. Using the task graphs of five tasks from EgoPER [4], *i.e.*, making *coffee*, *oatmeal*, *tea*, *pinwheels*, and *quesadilla*, we designed feasible multi-task transcripts that interweave two or three tasks within a single sequence. We do not include videos involving more than three tasks, as interleaving more than three recipes simultaneously is uncommon in real-life scenarios. During recording, participants received instructions for their next action through an earphone, ensuring that the environmental audio in the videos remained free of instructional speech. While our experiments utilized only the RGB video streams, the MEKA dataset includes multiple modalities: RGB, depth, audio, gaze, and hand-tracking data, offering valuable opportunities for future research in multi-modal multi-task temporal action segmentation.

1.2. Annotation and Statistics

We collected 100 multi-task egocentric videos, totaling approximately 12 hours of footage, with an average video length of seven minutes. Each video contains an average of 7.3 task switches. Leveraging the EgoPER dataset, we observed that some tasks share common actions, such as "transfer water to kettle" for both coffee and tea. To reduce redundancy, we merged these shared actions, resulting in 50 distinct action classes, along with one background class, for a total of 51 action classes. Each video in the MEKA dataset is fully annotated with frame-wise action labels.

Before annotation, annotators reviewed sample videos and corresponding annotations from the original EgoPER dataset to ensure consistency in labeling. However, we still observed discrepancies in the average durations for action segments, as shown in Fig. 1. While most action segments in MEKA are slightly shorter, a few actions, such as "slowly pour the rest of water in circular motion" and "microwave for X seconds", show significant differences. These actions often involve waiting time, which are reduced in the multi-task setting as participants tend to switch to other tasks during waiting times. Such discrepancies between the training and testing datasets pose additional challenges for training MT-TAS models using single-task videos.

Method	Acc	Acc-bg	Edit	F1@{10,25,50}		
Concatenation	65.7	51.3	53.9	59.1	56.3	46.3
Random Switch	66.6	61.2	58.8	62.9	60.6	50.9
LLM Switch	**67.8**	**63.7**	**64.2**	**68.9**	**66.8**	**56.0**

Table 1. Ablation studies on MSB.

1.3. Ethical Considerations

All participants provided informed consent for data collection and distribution. The dataset has been anonymized to protect participant privacy, and any identifiable information has been removed. The MEKA dataset will be made publicly available for research purposes upon publication.

2. Additional Implementation Details

We use GPT-4 [1] as the LLM to make task-switching decisions in the MSB module and to generate relevant objects for each action class in the DIVE approach. For video representations, we extract 2048-dimensional I3D features [2], pretrained on Kinetics, using a sliding window of 32 frames. The video frames are sampled at 10 fps. We generate the foreground and background frames by applying a Gaussian filter with a standard deviation of 20 to the images. In the FAAR module, we utilize the ViT-B/16 architecture of CLIP [6] to extract 512-dimensional image features from the foreground frames. We train the model with the MSTCN backbone for 200 epochs, with the ProTAS backbone for 100 epochs, and with the FACT backbone for 350 epochs. Each training process is equally divided into two stages: the first half focuses on training the model without FAAR, and the second half incorporates FAAR. For domain adaptation, we add a GRL before the output layer of the first stage of the model architecture. The domain classifier is a two-layer MLP with a hidden dimension of 64. The training time for the MSTCN and ProTAS backbones is approximately 3 hours, while the FACT backbone takes around 20 hours, all conducted on an NVIDIA RTX 6000 GPU.

3. Additional Results

Ablation Studies on MSB. In Table 1, we evaluate the effectiveness of the LLM's decisions on task switches in our MSB module by comparing it with two alternative methods: 1) *Concatenation*, where we simply concatenate videos of different tasks without interleaving actions; 2) *Random Switch*, where we use the same blending method but decide to continue or switch tasks randomly, without consulting the

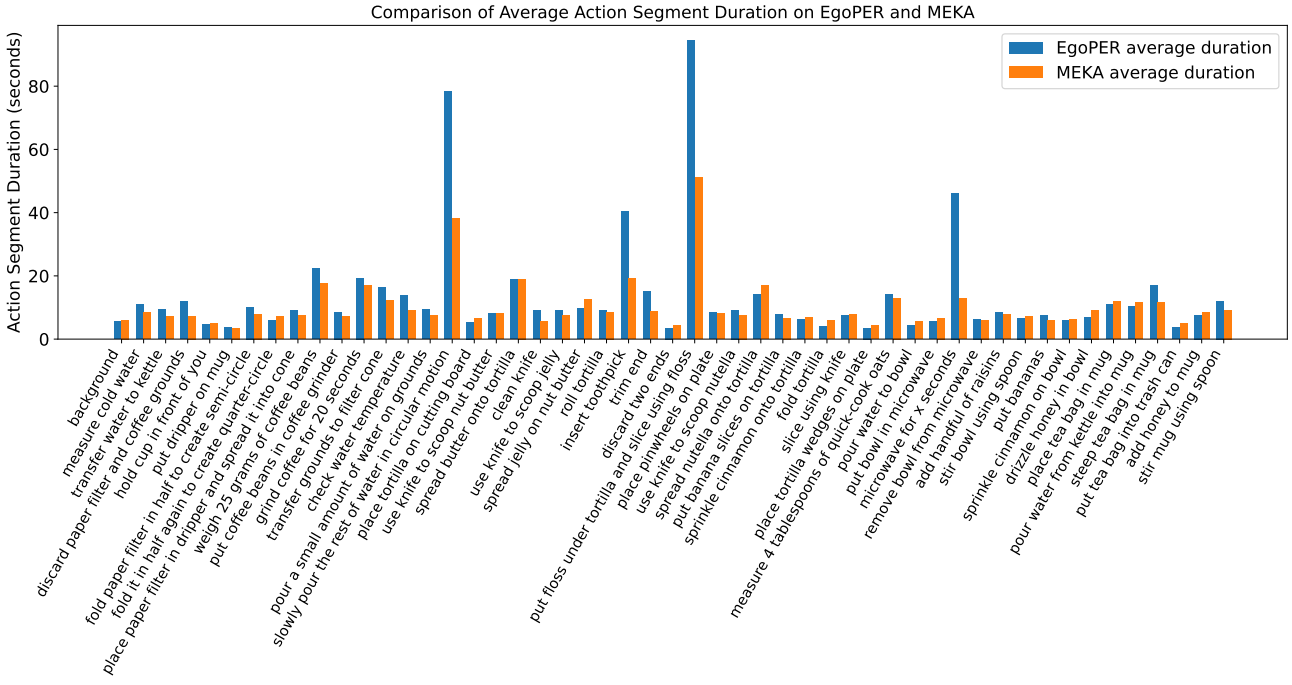

Figure 1. The average duration of action segments on EgoPER and MEKA.

LLM. The *Concatenation* method shows the lowest performance across all metrics, with an overall accuracy of 65.7% and F1@50 of 46.3%. This indicates that merely concatenating single-task videos without interleaving does not adequately expose the model to the complexities of multi-task scenarios, limiting its ability to generalize. The superior Edit and F1 scores of the *LLM Switch* method suggest that incorporating LLM decisions leads to more coherent and realistic multi-task sequences compared to random decisions.

Analysis on SBL Recovered Features. We evaluated the effectiveness of SBL using two analyses in Figure 2. We first computed pairwise cosine distances between adjacent frame features with and without SBL. The results show that SBL significantly reduces abrupt feature changes at task-switching points, leading to smoother transitions. Additionally, we visualized feature embeddings of boundary frames from two different tasks. Without SBL, the features exhibit a large separation, indicating a lack of temporal coherence. With SBL, the features of adjacent segments are brought closer together, suggesting that SBL effectively bridges the transition between tasks.

Ablation Study on DIVE. The DIVE module is a critical component of our framework, serving as the foundation for both FBFC and FAAR modules. It generates the necessary foreground and background frames for FBFC, and allows FAAR to focus on action-relevant regions. Without DIVE, FBFC becomes inapplicable, and FAAR must operate on full frames instead of foreground regions. To better understand DIVE's contribution, we conducted two additional ablation experiments: No DIVE during training: We removed FBFC and trained FAAR using full-frame inputs. During testing, we restored DIVE and provided foreground frames to FAAR. No DIVE during testing: We trained the full model with DIVE but removed it at inference time, feeding full images to FAAR. The results in Table 1 provide a comprehensive understanding of DIVE's importance in enabling foreground-aware learning and maintaining the robustness of our segmentation framework.

Method	Acc	Edit	F1@10	F1@25	F1@50
No DIVE in training	74.1	73.2	76.5	75.4	65.8
No DIVE in testing	71.7	68.5	72.6	71.1	61.7
w/ DIVE (Ours)	**75.7**	**74.9**	**79.7**	**77.6**	**67.4**

Table 2. Ablation study on the DIVE module. Removing DIVE during training or testing degrades performance.

Evaluation in Single-Task Settings. Although our approach is designed for multi-task temporal action segmentation (MT-TAS), we also assess its effectiveness in standard single-task scenarios by training and evaluating on the EgoPER dataset [4] using the MSTCN backbone. Interestingly, our method not only generalizes to multi-task settings but also enhances performance in single-task contexts. Specifically, it improves accuracy by 6.7% and F1@50 by 7.2% compared to the baseline, demonstrating the robustness and broad applicability of our framework.

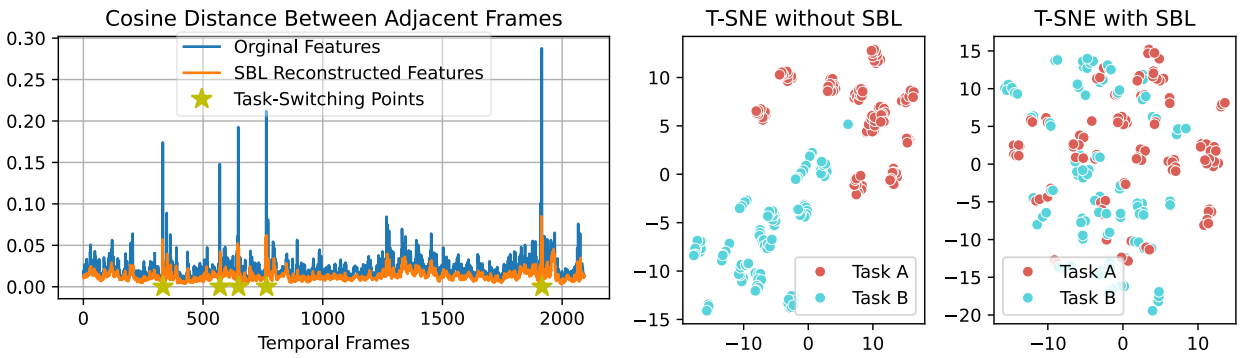

Figure 2. Analysis on SBL recovered features. Left: Feature distances between adjacent frames, showing that SBL smooths abrupt transitions at task switches. Right: T-SNE on feature embeddings of boundary frames from two tasks. Without SBL, features are widely separated; with SBL, features are drawn closer.

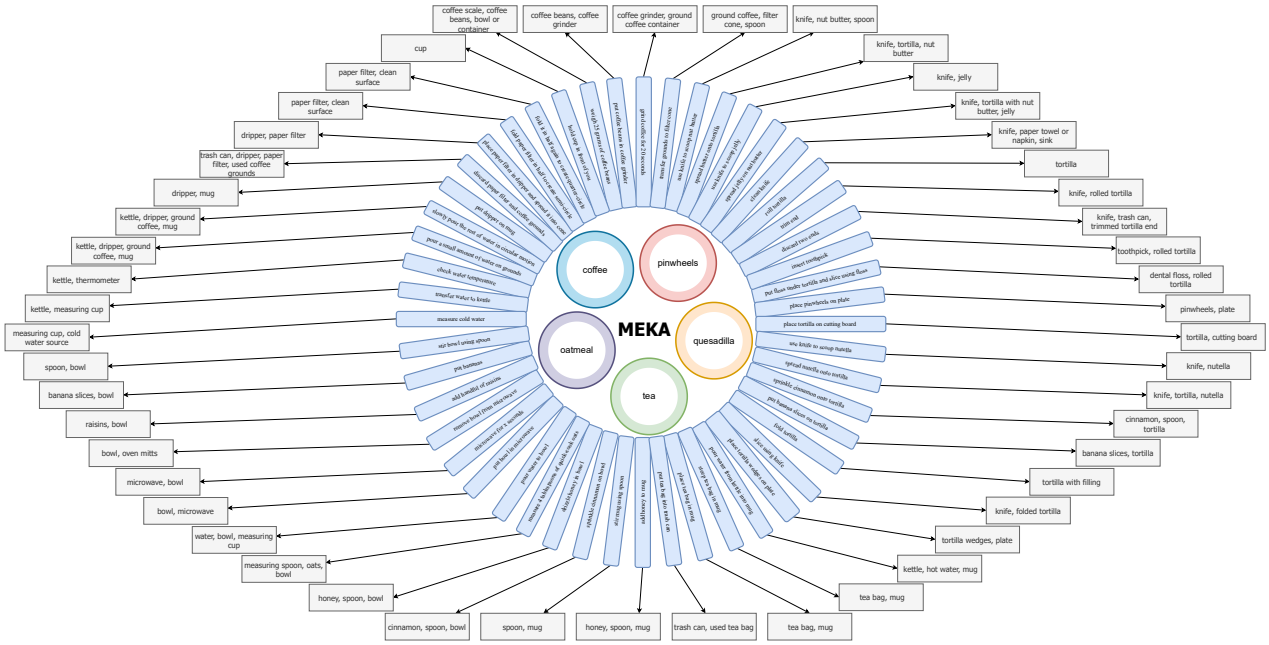

Figure 3. The relevant objects for each action class output by GPT-4, on the MEKA dataset.

Method	Acc	Edit	F1@10	F1@25	F1@50
baseline	84.7	94.2	93.1	91.6	86.1
ours	**92.4**	**97.9**	**96.4**	**96.4**	**93.3**

Table 3. Evaluation on single-task videos.

Relevant Objects Output by LLM. Fig. 3 illustrates the relevant objects identified by GPT-4 for each action class. Overall, the outputs are highly effective, accurately capturing the most critical objects needed for each action. For instance, GPT-4 successfully identifies relevant objects explicitly mentioned in the action descriptions, such as "spoon" and "bowl" for "stir bowl using spoon", and "dripper" and "mug" for "put dripper on mug". More importantly, GPT-4 goes beyond merely extracting nouns from action descriptions by including additional objects, such as "thermometer" for "check water temperature" and "knife" for "spread butter onto tortilla", demonstrating its ability to infer tools or objects that are implied rather than explicitly stated. This approach may introduce occasional noise as well, such as including an unnecessary tool "spoon" for "add honey to mug". Overall, the results showcase GPT-4's strong capability to identify and enrich relevant objects.

Qualitative Results. In Fig. 4, we visualize the results of our methods against two baseline models, ProTAS (online) and MSTCN (offline), on two test videos. In the first video, which involves making tea and pinwheels, the base-

line models struggle with task interleaving, frequently missing actions or misclassifying segments. In contrast, our offline model accurately detects nearly all actions, closely matching the ground truth and effectively capturing task interruptions and resumptions. The second video presents increased complexity with three concurrent tasks: making oatmeal, quesadilla, and tea. While the performance of the baseline models deteriorates further, our models maintain robust segmentation accuracy and correctly segment most actions. Although our online model tends to oversegment, it still handles task switches effectively. These results demonstrate the superior capability of our method in handling real-world multi-task scenarios in both online and offline settings.

Additionally, we present two qualitative visualizations in Figure 5. We display selected segments with three representative frames, and the ground-truth labels (GT) alongside our model's predictions (Pred), including the action class names, task labels, and corresponding start and end times (in seconds). The first example illustrates a successful case where our model not only predicts the correct action classes of different tasks but also achieves highly accurate temporal boundaries for each segment. The second example is a failure case in a multi-task video involving interleaved steps of making pinwheels and quesadilla. The model correctly identifies the step "put banana slices on tortilla" (quesadilla). However, during a subsequent step "insert toothpick" (pinwheels), it misclassifies the beginning and ending frames as "put banana slices on tortilla" (quesadilla). This confusion likely stems from the presence of another tortilla covered with banana slices, which occupies a significant portion of the frame and visually dominates the scene, misleading the model. Additionally, during the step "trim end" (pinwheels), the model incorrectly predicts "slice using knife", another step associated with making quesadilla. This misclassification is likely due to the presence of a folded tortilla and a knife. These errors underscore the need for *instance-aware recognition*, where the model can differentiate between multiple visually similar objects, such as two different tortillas used in separate tasks. Incorporating object instance disambiguation could substantially improve segmentation performance in complex multi-task scenarios.

4. Experiments on Adapted EGTEA Dataset

Adapted EGTEA Dataset. To further evaluate our approach, we adapted the EGTEA dataset [5] for MT-TAS evaluation. The EGTEA dataset originally contains 86 egocentric videos of seven different cooking recipes, including *BaconAndEggs*, *Cheeseburger*, *ContinentalBreakfast*, *GreekSalad*, *PastaSalad*, *Pizza*, and *TurkeySandwich*. We manually split each recipe video into two "tasks" by dividing 44 videos into two halves, with each half containing steps exclusively from one "task", effectively creating single-task videos. The remaining 42 videos include steps from two "tasks" presented in an interleaved manner, representing multi-task videos. As a result, we obtained a dataset comprising 88 single-task videos and 42 multi-task videos. This dataset contains 50 action classes and around 28 hours of video in total. We use the single-task videos for training and the multi-task videos for testing, aligning with the setup of our primary experiments.

MT-TAS Performance. Table 4 shows the results of both offline (using MSTCN [3] as the base model) and online (using ProTAS [7] as the base model) MT-TAS on the adapted EGTEA dataset. We observe that incorporating our proposed modules leads to progressive improvements in both offline and online MT-TAS settings. Specifically, in the offline setting, the accuracy increases from 82.6% in the baseline to 84.4% when all modules are applied, and the F1@50 score improves by 3.8%. In the online setting, the accuracy improves from 79.9% to 83.7%, with the F1@50 score increasing by 4.6%. Although the improvements on the adapted EGTEA dataset are not as remarkable as those observed on the MEKA dataset, they demonstrate that our modules effectively enhance performance even when the dataset does not exhibit extensive task interleaving. The smaller gains can be attributed to the nature of the adapted EGTEA dataset, where interleaving between "tasks" is less common, and the object layouts in single-task and multi-task videos are more similar. Nonetheless, the consistent improvements confirm the general applicability of our approach across different datasets and settings.

5. Complexity and Limitations

Complexity. Our proposed framework introduces specialized modules, including MSB, SBL, FBFC, and FAAR, that significantly enhance the model's adaptability to multi-task scenarios in temporal action segmentation. The MSB module synthesizes training data by blending single-task videos with pre-generated LLM queries, minimizing runtime impact as queries are processed offline. SBL and FBFC enhance feature representation through additional neural network layers, which may increase model size and computation cost during training. Notably, during inference, these modules (MSB, SBL, and FBFC) are inactive, so they do not add any computational complexity. Only the FAAR module remains active during inference, focusing on detecting action-relevant objects and extracting foreground features. Our ablation studies demonstrate that by limiting K, the number of top predicted actions considered, to 3, we effectively balance accuracy and inference efficiency.

Limitations and Future Work. Our framework effectively addresses critical challenges in multi-task temporal action segmentation, laying a strong foundation for future advancements in the field. Due to the lack of exist-

MSB	SBL	FBFC	FAAR	Offline MT-TAS [3] Acc	Acc-bg	Edit	F1@{10,25,50}			Online MT-TAS [7] Acc	Acc-bg	Edit	F1@{10,25,50}		
baseline				82.6	36.5	20.5	28.5	23.7	14.4	79.9	34.2	18.7	20.8	15.4	8.0
✓				83.1	37.4	24.2	32.6	27.5	16.2	81.5	33.2	23.9	23.4	18.0	8.5
✓	✓			82.8	39.5	25.8	33.8	27.5	16.4	82.4	35.8	24.1	23.1	18.5	9.1
✓	✓	✓		82.9	40.1	26.0	34.1	27.7	17.1	82.9	40.2	24.5	24.8	20.0	11.5
✓	✓	✓	✓	**84.4**	**44.2**	**28.8**	**35.5**	**30.3**	**18.2**	**83.7**	**46.0**	**27.4**	**26.1**	**21.0**	**12.6**

Table 4. Comparison of offline and online multi-task temporal action segmentation performance on the Adapted EGTEA dataset.

Figure 4. Qualitative visualization on multi-task temporal action segmentation results of multiple methods.

ing datasets for MT-TAS, we validated our approach primarily on our collected MEKA dataset and an auxiliary adapted EGTEA dataset. While these datasets provide a solid evaluation foundation on kitchen activities, extending our method to diverse domains like sports or industrial tasks would demonstrate broader generalizability and adaptability. Besides, our use of LLMs for deciding task switches and identifying relevant objects showcases the integration of advanced language models in action segmentation. In scenarios where LLMs are unavailable, alternative methods such as utilizing multi-task transcripts, leveraging knowledge bases, or employing action description parsing can be explored to achieve similar outcomes. Additionally, the DIVE approach benefits from the ongoing advancements in open-vocabulary object detection models. As these models continue to improve in efficiency and accuracy, we antici-

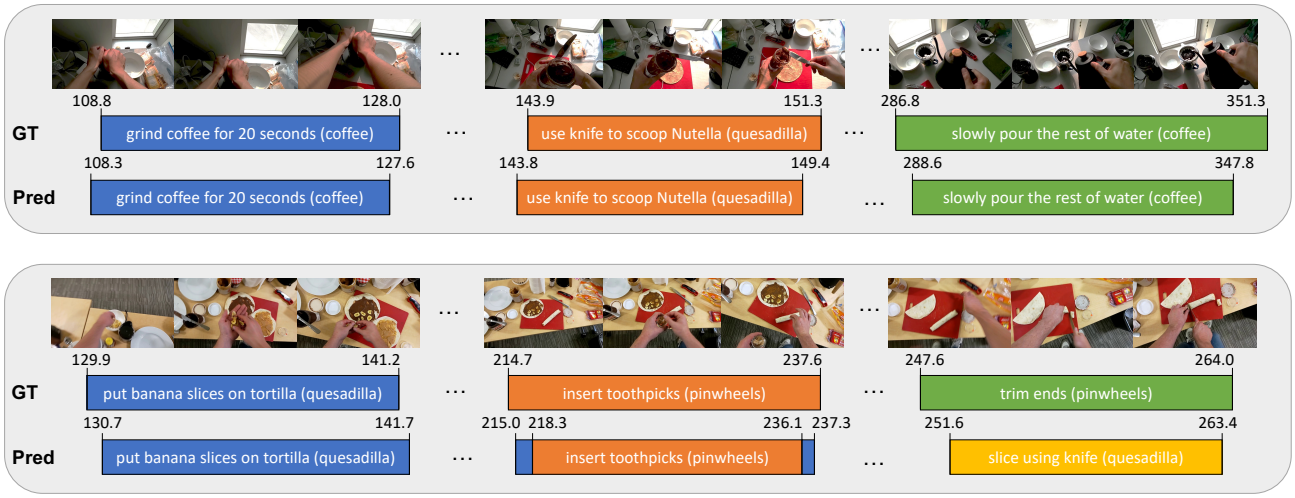

Figure 5. Visualization of our model's predictions on multi-task temporal action segmentation videos.

pate corresponding enhancements in our framework's performance. Lastly, while our domain adaptation strategy demonstrates performance improvements, our experiments show that there remains room for further improvement. Future work could explore more sophisticated adaptation techniques, such as action-level or video-level adaptation, to further mitigate the discrepancies between single-task and multi-task videos.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1

[3] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3575–3584, 2019. 4, 5

[4] S. Lee, Z. Lu, Z. Zhang, M. Hoai, and E. Elhamifar. Error detection in egocentric procedural task videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2

[5] Yin Li, Miao Liu, and James M. Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. *European Conference on Computer Vision*, 2018. 4

[6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *International Conference on Machine learning*, 2021. 1

[7] Y. Shen and E. Elhamifar. Progress-aware online action segmentation for egocentric procedural task videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 4, 5