Ego-METAS: an Egocentric online Multimodal Energy-efficient Temporal Action Segmentation benchmark

Maria Santos-Villafranca[1] **Jesus Bermudez-Cameo**[1] **Alejandro Perez-Yus**[1]
Giovanni Maria Farinella[2] **Antonino Furnari**[2]

[1]University of Zaragoza - I3A
[2]Department of Mathematics and Computer Science, University of Catania

Abstract

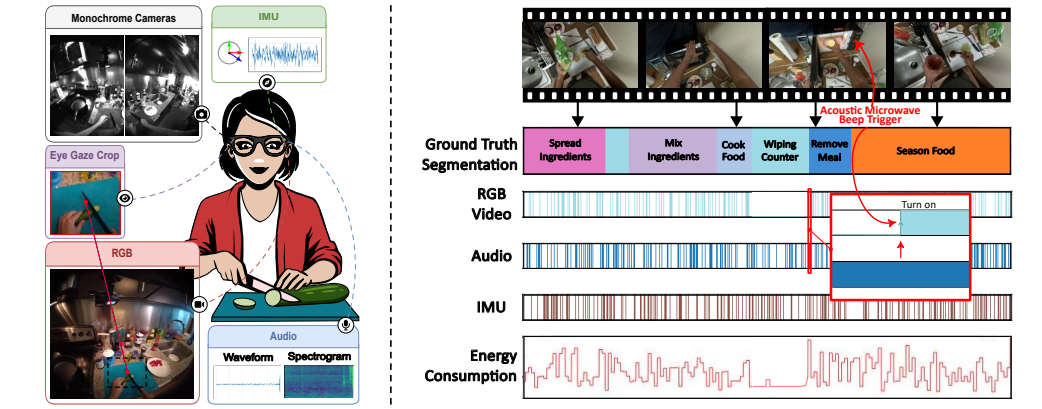
To operate in the physical world, embodied agents must perceive their environment in an "always-on" fashion, selectively accessing the most informative sensors to balance energy constraints and task accuracy. Despite its importance for resource-constrained devices, energy-aware perception remains under-explored, with most prior work assuming unlimited compute. To address this, we introduce **Ego-METAS**: the first **Ego**centric online **M**ultimodal **E**nergy-efficient **T**emporal **A**ction **S**egmentation benchmark. Ego-METAS provides a unified testbed of more than 100 hours of untrimmed egocentric video from EgoExo4D, CMU-MMAC, and CaptainCook4D, spanning 5 modalities (RGB, audio, gaze, IMU, and monochrome camera). We formulate an online temporal action segmentation task where models must dynamically select which sensors to activate at each timestep while strictly adhering to hardware-representative energy budgets. Alongside the benchmark, we release unified splits, cleaned annotations, pre-extracted features, and a diverse suite of baseline routing policies. Our evaluations show that optimal routing is highly scenario-dependent, and that existing policy-learning methods—designed primarily for trimmed clips—struggle to adapt to continuous, untrimmed environments. However, even simple dynamic fusion of complementary modalities (e.g., via random routing) proves critical for balancing predictive accuracy against strict energy budgets. Ultimately, Ego-METAS provides a standardized foundation to develop robust, cost-aware policies for autonomous, always-on embodied AI.

1 Introduction

Continuous perception is essential for intelligent agents operating in the physical world, such as robots or assistants deployed through smart glasses [36]. Yet, modern computer vision systems often treat perception as exhaustive, processing all available modalities under implicit assumptions of abundant energy, computation, and memory. This contrasts with biological perception, where multimodal sensing is used selectively to cope with physical constraints: for instance, audio may suffice during a quiet conversation while washing dishes, whereas in a crowded bar we naturally rely more on vision, such as lip reading, to communicate effectively.

We argue that practical embodied AI should reconsider this scaling assumption: rather than treating physical constraints as a nuisance to be overcome by better hardware, we should design systems that embrace resource budgeting and selective modality routing as core algorithmic features. As illustrated in Fig. 1, wearable devices are inherently multi-modal. In these settings, an assistive system could efficiently recognize actions by keeping its energy-intensive RGB camera active while a user puts a meal in the microwave, but power it down during a prolonged, repetitive background task like wiping

Preprint.



(a) Egocentric Sensing is Inherently Multimodal (b) Proposed Multimodal Energy-efficient Temporal Action Segmentation Setup

Figure 1: (a) Wearable devices naturally capture rich streams including multiple modalities. (b) In the proposed setup, models dynamically select which sensory modalities to activate for the next timestep, minimizing energy consumption while maintaining accurate, continuous online action segmentation.

the counter. During this time, the system can rely entirely on low-power audio and IMU sensors until an acoustic trigger—such as the microwave beep—dynamically reactivates the visual stream. Current approaches have largely overlooked this dynamic, online setting due to the absence of unified, standardized benchmarks tailored for energy-aware, continuous processing. Indeed, most prior efforts in Temporal Action Segmentation (TAS) focused on static uni-modal settings [13], or exhaustive multi-modal processing [39], with limited works focusing on energy-aware scenarios [21, 34, 51], often scattered and carried out under non-standardized benchmarks.

To address this gap, we introduce EgoMETAS: an Egocentric online Multimodal Energy-efficient Temporal Action Segmentation benchmark. Specifically, Ego-METAS introduces a unified evaluation pipeline designed specifically to test systems under strict operational constraints. We source egocentric videos from three distinct egocentric datasets, EgoExo4D [21], CaptainCook4D [35], and CMU-MMAC [12], which are selected to be representative of real-world operational settings, having been collected natively with diverse hardware such as Aria Glasses, Microsoft HoloLens, and custom multimodal rigs. Unlike prior efforts that restrict evaluation to episodic recognition on single devices [21], Ego-METAS is the first benchmark to target untrimmed, continuous action segmentation across diverse wearable platforms. To resolve historical inconsistencies in prior datasets, we rigorously revise existing annotations and establish a unified evaluation protocol that moves beyond standard task accuracy, allowing us to quantitatively assess model performance against explicit energy consumption limits and strict operational budgets. Furthermore, to accurately reflect the physical realities of wearable systems, we curate a principled set of modalities, ranging from full RGB streams to energy-efficient grayscale images and gaze-centered crops, and provide concrete estimates for their respective acquisition, memory, and processing costs. Alongside this benchmark, we systematically survey the literature on resource-constrained perception and compare the most viable approaches. We open-source unified splits, cleaned annotations, pre-extracted modality features, and evaluation scripts, providing a robust foundation for the community to build upon. Ultimately, Ego-METAS shifts the focus toward designing systems that embrace hardware and energy constraints as core features rather than limitations, laying the groundwork for truly viable, always-on embodied AI.

2 Related Work

Energy Efficient Methods: Deploying vision systems on low-power devices [5, 46] typically relies on static network optimizations [5, 19, 50]. Standard approaches include quantization [8, 14, 19, 49], pruning [1, 55], and compact architectures [17, 25]. Additional overhead reductions are achieved through Neural Architecture Search [46, 57], efficient recurrent networks like Mamba [22] and xLSTM [2], and knowledge distillation [23]—applied specifically to egocentric perception by EgoDistill [47] to compress video and IMU semantics. However, static model optimization is inherently insufficient for continuous, in-the-wild sensing. To achieve true always-on processing for resource-constrained wearables, systems must move beyond fixed computation and learn dynamic routing policies that dynamically adapt the active sensory budget on the fly.

Information Selection: Beyond rigid uniform sampling [52], efficient video understanding relies on extracting salient temporal context. While query-based frame selectors [24] require offline video access, online methods successfully reduce temporal redundancy via lightweight clip sampling [28] or early-exit inference gates [18]. Ego-METAS elevates this paradigm: rather than merely selecting unimodal temporal frames, we formulate the problem as dynamic multimodal sensor routing, evaluating policies tailored for continuous, always-on temporal action segmentation.

Multimodal Learning: Multisensory integration significantly enhances representation learning [37, 38] but inherently multiplies computational costs. To mitigate this, AdaMML [34] introduced differentiable modality routing via the Gumbel-Softmax trick [26]. This foundational mechanism has been widely adapted [9, 53] for active feature prioritization [45], task-specific sensor gating [54], and cross-modal policy distillation [6]. Concurrent efforts address joint modality training [29], modality selection under domain shifts [31], and test-time missing modalities [40]. However, these approaches overwhelmingly evaluate on offline, pre-trimmed clips. Real-world continuous perception cannot rely on predefined action boundaries to dictate modality switching. To bridge this gap, Ego-METAS introduces a systematic framework specifically designed to evaluate Energy-Efficient Multimodal Online Temporal Action Segmentation in an always-on, untrimmed setting.

Temporal Action Segmentation (TAS): Offline TAS assumes full video access for frame-wise classification [13, 16, 30, 44]. Conversely, embodied wearables require causal, online TAS [42, 56], a setting that remains under-explored [13] despite foundational [11] and recent advances based on causal convolutions and memory banks [4, 42, 56]. We adopt online TAS as the foundation for Ego-METAS, challenging models to perform continuous segmentation under strict energy budgets via dynamic modality selection. While prior energy-efficient benchmarks like Ego-Exo4D [21] evaluate streaming keystep recognition on isolated episodes—explicitly filtering out long background sequences—Ego-METAS forces models to persistently navigate long, unsegmented background transitions in the wild. Furthermore, we expand beyond single-device constraints to establish a comprehensive multi-device testbed unifying Ego-Exo4D [21], CMU-MMAC [12], and CaptainCook4D [35].

3 Benchmark

We source egocentric videos and annotations from three existing datasets, namely CMU-MMAC [12], Ego-Exo4D [21], and CaptainCook4D [35], which have been collected using diverse hardware representative of the variety of multimodal observations which can be obtained with wearable devices. We provide a formal definition of METAS and a principled set of evaluation metrics aimed to assess both segmentation accuracy, energy efficiency, and their trade-offs.

3.1 Task definition

Let $\mathcal{X}_{:t} = \{\mathbf{X}_{:t}^{(m)}\}_{m=1}^{M}$ denote the continuous input stream up to time t from M sensors (e.g., RGB, audio, IMU). Despite varying native acquisition frequencies, we define a common discrete clock t for system predictions. Operating under a strict hardware power budget B (in mW), the online inference pipeline at step t proceeds in three stages: 1) A **routing policy** π, conditioned on historically realized features $\{\tilde{\mathbf{x}}_{:t-1}^{(m)}\}_m$, outputs a binary decision vector $\mathbf{a}_t \in \{0,1\}^M$. If $a_t^{(m)} = 1$, sensor m actively collects observation $\mathbf{X}_t^{(m)}$, incurring capture cost $E_{\text{cap}}^{(m)}$. 2) Active sensors utilize feature extractors ϕ_m to generate dense features $\mathbf{x}_t^{(m)} = \phi_m(\mathbf{X}_t^{(m)})$. Inactive sensors bypass extraction, relying on a placeholder $\mathbf{z}_t^{(m)}$ (we set this to the previous feature $\tilde{\mathbf{x}}_{t-1}^{(m)}$) This yields a strictly defined effective representation: $\tilde{\mathbf{x}}_t^{(m)} = a_t^{(m)}\mathbf{x}_t^{(m)} + (1 - a_t^{(m)})\mathbf{z}_t^{(m)}$. 3) The effective history $\{\tilde{\mathbf{x}}_{:t}^{(m)}\}_m$ is processed by a **TAS model** Ψ to predict the current action class $\hat{y}_t$. Alongside capture costs, feature extraction and TAS computation incur specific processing ($E_{\text{comp}}^{(m)}$) and memory access ($E_{\text{mem}}^{(m)}$) costs. The objective of Ego-METAS is to learn a joint system (π, Ψ) that maximizes prediction accuracy ($\hat{y}_t = y_t, \forall t$) subject to the constraint that total aggregated energy ($E_{\text{cap}} + E_{\text{comp}} + E_{\text{mem}}$) across all active components never exceeds budget B at any timestep t.

3.2 Energy computation

For video $\mathcal{V}$ of duration T, total energy (mJ) is the sum of capture, computation and memory energy:

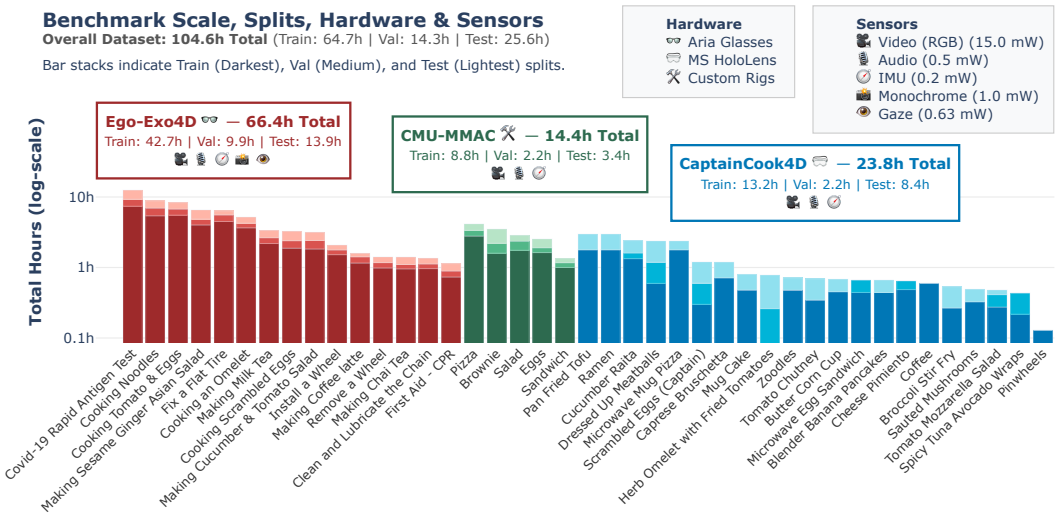


Figure 2: Ego-METAS comprises 104.6 hours of video and 41 scenarios collected with three different wearable devices, encompassing 5 modalities captured with varying energy consumption profiles.

$$E_{\mathcal{V}_{\text{total}}} = \sum_{t=1}^{T} \Bigg(\underbrace{\sum_{m=1}^{M} a_t^{(m)} P_m \Delta t}_{E_{\mathcal{V}_{\text{cap}}}} + \underbrace{\alpha \left(f_{\Psi}^{\text{MAC}} + f_{\pi}^{\text{MAC}} + \sum_{m=1}^{M} a_t^{(m)} f_{\phi_m}^{\text{MAC}} \right)}_{E_{\mathcal{V}_{\text{comp}}}} + \underbrace{\beta \left(f_{\Psi}^{\text{MEM}} + f_{\pi}^{\text{MEM}} + \sum_{m=1}^{M} a_t^{(m)} f_{\phi_m}^{\text{MEM}} \right)}_{E_{\mathcal{V}_{\text{mem}}}} \Bigg) \tag{1}$$

where P_m is the power consumption of sensor m, Δt is the temporal duration between consecutive timesteps. The terms f_{Ψ}^{MAC}, f_{π}^{MAC}, f_{Ψ}^{MEM}, f_{π}^{MEM} represent the MAC operations and CUDA memory events of the core TAS model Ψ and policy π (we set these to zero for non-learned policies), while $f_{\phi_m}^{\text{MAC}}$ and $f_{\phi_m}^{\text{MEM}}$ denote the corresponding costs for the feature extractor ϕ_m. The constants α and β convert these operations to energy (mJ) [21]. The above formulas allow to compute the total energy for a whole video. In practice, we also compute the power $P_{\mathcal{V}} = \frac{E_{\mathcal{V}_{\text{total}}}}{T}$ as the total energy divided by the length of the video in seconds T (in mW). In our evaluations, we constrain our models to operate within fixed energy budgets B, which is achieved by modifying the model's configuration so that $P_{\mathcal{V}} < B$. For more details of the estimation of the energy please refer to Sec. A.4.

3.3 Datasets, Modalities and Splits

Ego-METAS comprises 104.6 hours of egocentric video across 41 scenarios. Sourced from three established multimodal datasets described below, the benchmark covers diverse hardware profiles and five sensory modalities with distinct energy signatures, including standardized splits (see Figure 2).

Ego-Exo4D [21]: Captured via Aria glasses, this dataset provides procedural activities across five modalities (RGB, audio, IMU, monochrome, gaze). Because prior benchmarks evaluated isolated episodes, the original keystep annotations exhibit inconsistent granularity and overlapping classes that preclude continuous perception (Sec. A.1). To resolve this, we introduce a rigorous re-annotation of this subset, enabling robust always-on evaluation (Sec. A.2). This dataset introduces the most diverse array of scenarios outside of the cooking domain and constitutes 66 h of the total benchmark.

CMU-MMAC [12]: Recorded using custom multimodal rigs, captures subjects performing kitchen tasks in a lab setting. It spans five different recipes, with multiple modalities, from which we select video, audio, and the accelerometer and gyroscopes from the IMUs, accounting to 16 hours.

CaptainCook4D [35]: Captured via Microsoft HoloLens, this dataset focuses on step-by-step procedural execution and mistake detection. Original annotations provided 325 different classes

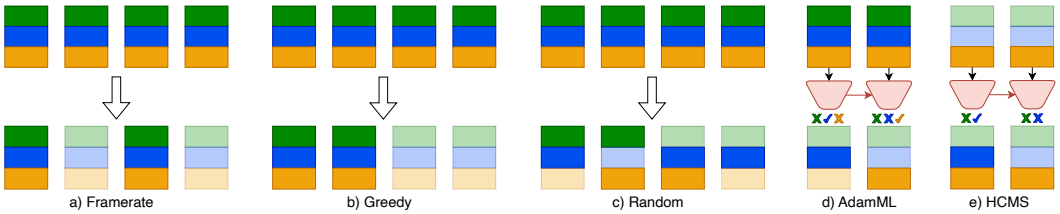


Figure 3: **Policies**. a) Framerate drops entire frames at fixed intervals; b) Greedy uses all modalities until the budget is consumed; c) Random drops individual modalities with a given probability; d-e) Learned policies predict which sensors should be active at each step.

which was too granular for the task we are proposing. Thus, we re-annotated as described in Sec. A.2. The dataset involves 21 different cooking recipes, with synchronized video, audio, and IMU streams.

3.4 Pre-Extracted Features

As part of the dataset, we provide the extracted features from each modality. Particularly, a feature vector $\mathbf{x}_t^{(m)}$ is pre-extracted per modality m at every timestep t. We choose a temporal discretization of 30 fps following video. Therefore, all modalities are synchronized with a common time frame. For each modality, we use frozen feature extractors from the literature. Details are reported in Sec. A.7.

3.5 Evaluation protocol and metrics

Models are required to produce predictions at 30 FPS (possibly replicating old predictions when idle), even when trained at lower temporal resolutions. This ensures a consistent evaluation setting and prevents models from simplifying the task by skipping frames or actions. Furthermore, following the energy constraints proposed in [21], we define two operating regimes per second for the METAS task: a low-power budget of 20 mW and a high-performance budget of 2.8 W. These budgets are real-device oriented. Tentatively, low-budget settings targets smaller wearable devices (such as Ray-Ban Meta smart glasses [32]), which require low power consumption in order to last for a full working day. In contrast, the Therefore, these budgets are designed to mimic real-world scenarios and encourage models to be not only accurate but also energy-efficient for deployment on real devices.

Following standard TAS literature [13], we evaluate segmentation accuracy using Frame Accuracy (Acc), segmental Edit Distance (Edit) to measure the sequential ordering of predicted actions, and Segmental F1 Scores at varying temporal Intersection-over-Union thresholds (F1@10, 25, 50) to penalize over-segmentation. To evaluate efficiency, we report these metrics alongside the average Energy Consumption per second (mW), and percentage of times a given modality is sensed across the video. While previous works tackling episodic keystep detection [21] used ranking-based metrics like mean Average Precision (mcAP), we argue that these are structurally inadequate for continuous Temporal Action Segmentation, as detailed in Sec. A.1.

4 Baseline Policies

To evaluate the proposed benchmark, we use ProTAS [41] as the state-of-the-art model to perform the temporal action segmentation of the videos. Using it as a backbone TAS architecture, we consider two families of policies. Fixed policies use heuristics to operate under a fixed energy budget, whereas learned policies involve learning an auxiliary network to decide which modalities to activate at each timestep, often based on a low-power modality input. Specifically, we evaluate three fixed policies and two learned ones, adapted from the literature (Figure 3).

Frame Rate [21]: This policy (Fig. 3 (a)) reduces the video sampling frequency to save energy. For skipped timesteps, the model outputs the most recent prediction. While decreasing the framerate significantly reduces the energy consumption, it may result in the loss of fine-grained temporal resolution, and short-duration actions may be entirely missed due to temporal aliasing.

Greedy [21]: This policy (Fig. 3 (b)) activates all modalities until reaching the predetermined energy budget per second. When the next second starts, it reactivates all modalities again, repeating the cycle.

Table 1: Performance on Ego-Exo4D [21] across energy budgets and policies. For learned policies, split cells report the modality usage of the TAS model (left) and the policy model (right).

Budget	#	Policy	Parameters						Energy	Acc	mAP	Edit	F1@10	F1@25	F1@50
No Budget	1	Frame Rate	30.00 FPS	100	100	✗	100	✗	11.53 W	45.08	33.06	23.86	21.74	17.85	10.46
2.8 W	2	Frame Rate	06.00 FPS	20.0	✗	✗	✗	✗	01.86 W	36.47	22.76	19.04	19.98	15.81	8.10
	3	Frame Rate	10.00 FPS	✗	33.3	✗	✗	✗	00.14 W	25.08	14.95	13.06	12.77	7.28	2.97
	4	Frame Rate	30.00 FPS	✗	✗	100	✗	✗	00.00 W	24.80	13.92	13.17	12.56	8.44	2.92
	5	Frame Rate	30.00 FPS	✗	✗	✗	100	✗	01.82 W	34.58	21.79	14.29	13.52	10.58	5.89
	6	Frame Rate	15.00 FPS	✗	✗	✗	✗	50.0	01.54 W	24.48	14.94	10.96	10.69	7.58	3.87
	7	Frame Rate	06.00 FPS	20.0	20.0	✗	✗	✗	01.94 W	26.02	15.71	11.54	12.39	8.07	3.74
	8	Frame Rate	30.00 FPS	✗	100	100	✗	✗	00.42 W	26.67	15.21	12.59	12.70	9.09	4.07
	9	Frame Rate	06.00 FPS	✗	✗	✗	20.0	20.0	00.98 W	43.71	26.82	25.34	24.62	19.05	12.02
	10	Frame Rate	05.00 FPS	16.7	16.7	16.7	✗	✗	01.62 W	38.03	23.95	25.13	24.06	18.55	10.90
	11	Frame Rate	06.00 FPS	20.0	20.0	✗	20.0	✗	02.31 W	45.99	33.56	26.58	25.22	20.96	10.61
	12	Greedy		13.4	13.4	13.4	13.4	13.4	02.37 W	43.75	31.65	25.32	25.83	20.96	12.30
	13	Random	τ_{tr}0.70, τ_{inf}0.900, c=0	10.1	10.0	10.0	10.0	10.0	01.78 W	47.02	30.55	26.37	25.61	20.07	12.93
	14	Random	τ_{tr}0.70, τ_{inf}0.900, c=1	5.1	11.6	20.5	8.5	7.4	01.14 W	46.80	30.47	26.51	25.91	20.18	12.80
	15	Random	τ_{tr}0.90, τ_{inf}0.900, c=1	5.1	11.5	20.5	8.5	7.4	01.13 W	45.30	29.75	26.70	25.53	20.57	11.90
	16	Random	τ_{tr}0.99, τ_{inf}0.900, c=1	5.1	11.5	20.5	8.5	7.4	01.14 W	38.26	26.44	22.24	21.73	17.77	10.19
	17	AdaMML		0.4	0.5 / 100	0.6 / 100	100	✗	02.25 W	22.10	12.11	12.73	10.81	6.29	2.60
	18	AdaMML		11.2	13.2 / 100	23.5 / 100	100	✗	03.26 W	27.67	16.44	17.37	15.88	11.58	5.51
	19	HCMS		15.0	37.7	100	27.8	19.4	03.35 W	40.91	27.67	22.83	21.74	17.17	11.24
	20	HCMS		0	0.0	100	0.0	0	00.01 W	17.82	13.89	15.57	9.84	6.26	3.03
20 mW	21	Frame Rate	00.05 FPS	0.2	✗	✗	✗	✗	16.29 mW	26.05	15.59	10.39	12.29	8.46	4.24
	22	Frame Rate	00.10 FPS	✗	✗	✗	0.3	0.3	16.70 mW	28.41	18.36	11.28	13.91	9.77	4.66
	23	Frame Rate	00.10 FPS	✗	0.3	0.3	0.3	0.3	18.13 mW	27.02	18.36	10.06	12.60	9.06	4.16
	24	Greedy		0	0	0	0	0	00.47 mW	22.44	12.42	7.06	9.78	6.54	2.17
	25	Random	τ_{tr}0.90, τ_{inf}0.997, c=1	0.2	0.3	0.6	0.3	0.2	34.93 mW	33.09	20.06	26.73	21.24	15.05	6.77
	26	Random	τ_{tr}0.90, τ_{inf}1.000, c=1	0	0	0	0	0	00.47 mW	12.00	11.60	2.25	0.53	0.18	0.15
	27	AdaMML		0	0.0 / 100	0.0 / 100	100	✗	2212 mW	18.13	12.29	5.43	5.83	2.92	0.95
	28	HCMS		0	0.0	100	0.0	0	05.00 mW	17.82	13.89	15.57	9.84	6.26	3.03

Sensors: Video (RGB), Audio, IMU, Monochrome, Gaze.

While easily adaptable for a fixed budget, this policy is inherently problematic for online TAS, as an exhausted budget midway through a sequence may preclude the model from capturing subsequent steps.

Random [21]: This policy (Fig. 3 (c)) randomly drops τ percentage of each modality. If all modalities are dropped at the same timestep, feature extraction is bypassed, but the TAS model still executes a forward pass, incurring the corresponding computational cost. We train this policy ad different dropout rates (τ_{train}) and evaluate the performance varying dropout at inference time (τ_{inf}). We report a plain ($c = 0$) and a cost-aware ($c = 1$) version, which biases selection prioritizing low-power modalities (details at Sec. A.6).

AdaMML [34]: This policy (3 (d) left) learns to select which modalities to use per timestep. For this policy, we take their original implementation and adapt its loss to handle our TAS setting. We use low-energy counterparts of RGB video as input to this policy to reduce computational overhead. Specifically, we use gaze crops for Ego-Exo4D [21], and the low-resolution version of RGB (192×192) for the rest of the datasets.

HCMS [51]: This policy (3 (d) right) sorts modalities by computational cost, and chooses the cheapest modality to be always on. A gating module decides whether to subsequently activate the remaining inactive modalities or not, ordered by cost. Therefore, the most expensive modality is never active on its own.

5 Experiments

We establish the first performance reference for the Multimodal Energy-efficient online Temporal Action Segmentation (METAS) task by discussing baseline performance, energy-accuracy trade-offs, and qualitative examples. Detailed experimental settings are reported in Sec. A.5.

5.1 Baseline comparison

Ego-Exo4D: Table 1 summarizes performance on Ego-Exo4D [21]. To establish an upper bound, the best unconstrained baseline at 30 FPS (row 1) achieves 45.08% accuracy but consumes a massive 11.53 W. Under the 2.8 W budget, static Frame Rate policies demonstrate that combining complementary modalities surpasses isolated sensors. While fusing the two strongest individual modalities (RGB and audio) unexpectedly degrades performance (row 7), pairing lightweight signals like gaze and monochrome (row 9) yields an efficient balance. The best static baseline (RGB, audio,

Table 2: Performance on CMU [12] across energy budgets and policies. For learned policies, split cells report the modality usage of the TAS model (left) and the policy model (right).

Budget	#	Policy	Parameters				Energy	Acc	mAP	Edit	F1@10	F1@25	F1@50
No Budget	1	Frame Rate	30.00 FPS	100.0	100.0	✗	09.72 W	85.63	53.73	30.41	35.93	32.77	26.18
2.8 W	2	Frame Rate	06.00 FPS	20.0	✗	✗	01.91 W	84.34	50.97	39.07	42.90	40.72	33.47
	3	Frame Rate	01.00 FPS	3.3	3.3	✗	00.32 W	83.17	44.23	44.75	46.95	43.34	37.12
	4	Frame Rate	01.00 FPS	3.3	3.3	3.3	00.32 W	82.47	43.79	40.57	44.14	40.17	33.26
	5	Greedy		26.7	26.7	26.7	02.60 W	79.49	29.38	27.08	25.95	24.48	19.92
	6	Random	τ_{tr}0.70, τ_{inf}0.900, c=0	10.1	10.1	10.0	00.98 W	80.97	39.30	28.34	30.35	27.82	24.95
	7	AdaMML		0.0 / 100	0.2 / 100	5.3 / 100	02.21 W	66.71	18.69	20.62	16.91	11.38	5.50
	8	HCMS		25.0	67.4	100.0	02.61 W	83.94	47.44	30.06	34.27	32.53	28.20
20 mW	9	Frame Rate	00.05 FPS	0.2	✗	✗	16.05 mW	69.37	21.54	24.15	28.18	23.80	14.79
	10	Frame Rate	00.05 FPS	0.2	0.2	✗	16.76 mW	70.27	23.00	24.62	29.18	24.94	15.12
	11	Frame Rate	00.05 FPS	0.2	0.2	0.2	16.77 mW	70.84	22.86	24.20	29.09	26.80	17.38
	12	Greedy		0	0	0	00.41 mW	66.26	5.34	0.00	0.00	0.00	0.00
	13	Random	τ_{tr}0.97, τ_{inf}1.000, c=0	0	0	0	00.40 mW	66.26	5.86	0.00	0.00	0.00	0.00
	14	AdaMML		0.0 / 100	0.2 / 100	5.3 / 100	2213 mW	66.71	18.69	20.62	16.91	11.38	5.50
	15	HCMS		0	0	100.0	05.02 mW	42.80	12.78	13.24	10.62	6.72	3.68

Sensors: Video (RGB), Audio, IMU, Monochrome, Gaze.

gaze; row 11) reaches 45.99% accuracy. A uniform Random policy ($c = 0$, row 13) proves superior, surpassing the 11.53 W upper bound using only 1.78 W. Furthermore, a cost-aware Random policy ($c = 1$, row 14) shifts usage toward cheaper sensors like IMU (20.5%) over RGB (5.1%), maintaining 46.80% accuracy while slashing energy to 1.14 W. While robust to high training dropouts, Random degrades at extreme values (rows 15-16). In contrast, complex learned policies like AdaMML (rows 17-18) struggle: the overhead of maintaining always-on modalities (gaze, audio, IMU) for routing decisions forces the model to sense other inputs $< 1\%$ of the time, crippling performance. HCMS (row 19) achieves strong accuracy but lags behind Random in efficiency; its hierarchical design forces all cheaper modalities to remain active merely to trigger RGB. Ultimately, as static policies confirm, naively activating all complementary sensors is often detrimental compared to dynamic, sparse sampling.

Under the extreme 20 mW budget, system dynamics invert. To adhere to the budget, static Frame Rate policies must sample rarely (e.g., RGB active 0.2% of the time in row 21). Yet, lightweight combinations like gaze and monochrome (row 22) still extract enough context to achieve 28.41% accuracy. This extreme constraint completely breaks discrete dynamic models. To avoid exceeding 20 mW, the Greedy policy (row 24) operates entirely blind (0% usage). Cost-aware Random policies ($c = 1$) are highly sensitive at this scale, either heavily violating the budget (row 25) or collapsing to 0% usage (row 26). AdaMML (row 27) fails fundamentally; its intrinsic architectural overhead exceeds 2.2 W even when no inputs are sensed, rendering it structurally unsuited for ultra-low power applications. Meanwhile, HCMS (row 28) adapts to the strict budget by devolving to use only IMU.

CMU-MMAC: Table 2 highlights how different dataset dynamics impact optimal routing. Unlike Ego-Exo4D, high frame rates on CMU-MMAC cause severe over-segmentation; thus, the unconstrained 30 FPS baseline (row 1) suffers a low edit score (30.41) despite consuming 9.72 W. Instead, sparse sampling acts as a natural regularizer, allowing static policies to strictly dominate. Under the 2.8 W budget, a 6 FPS RGB baseline (row 2) performs strongly, but dropping to 1 FPS and adding audio (row 3) maximizes the edit score (44.75) using just 0.32 W, slightly edging out full IMU fusion (row 4). Although evaluated at 30 FPS, processing at a lower frame rate implicitly simplifies the temporal task: by holding predictions constant across intermediate frames, the model avoids noisy temporal boundaries. This is especially beneficial in smooth scenarios where repeated predictions remain valid for longer periods. Consequently, dynamic routing models lag significantly, with Greedy (row 5) and Random (row 6) achieving substantially lower accuracy.

Under the extreme 20 mW constraint, dynamic policies collapse entirely to 0% modality usage (rows 12, 13), effectively operating blind by merely predicting the most common action sequence prior. In contrast, extreme sparse static sampling (0.05 FPS) successfully extracts sufficient context—whether using RGB alone, RGB and audio, or full IMU fusion (rows 9–11)—peaking at 70.84% accuracy. Finally, learned policies exhibit severe structural flaws in this regime. AdaMML's intrinsic overhead triggers massive energy violations, consuming between 2.2 W and 6.8 W regardless of actual sensor usage, rendering it fundamentally unsuited for ultra-low power applications (row 14). While HCMS (row 15) performs reasonably well under the 2.8 W budget, it is outcompeted by naive

Table 3: Performance on CaptainCook4D [35] across energy budgets and policies. Split cells report the modality usage of the TAS model (left) and the policy model (right).

Budget	#	Policy	Parameters	📹	🎙	🧭	Energy	Acc	mAP	Edit	F1@10	F1@25	F1@50
No Budget	1	Frame Rate	30.00 FPS	100.0	✗	✗	09.30 W	44.88	19.39	18.05	18.64	14.82	7.51
2.8 W	2	Frame Rate	01.00 FPS	3.3	✗	✗	00.31 W	43.23	18.45	36.48	32.62	28.88	14.08
	3	Frame Rate	00.50 FPS	1.7	1.7	✗	00.16 W	42.22	18.05	36.17	31.91	23.79	10.32
	4	Frame Rate	00.10 FPS	0.3	✗	0.3	00.03 W	39.23	17.31	35.53	31.10	23.68	14.21
	5	Frame Rate	00.50 FPS	1.6	1.6	1.6	00.16 W	40.62	17.47	37.01	29.81	22.62	11.54
	6	Greedy		26.1	26.1	✗	02.54 W	33.80	13.14	21.04	22.18	18.24	9.45
	7	Random	τ_{tr}0.97, τ_{inf}0.970, c=0	2.9	2.9	3.0	00.29 W	29.74	10.84	25.58	24.96	20.87	12.44
	8	AdaMML		1.4 / 100	1.4 / 100	1.4 / 100	02.30 W	22.60	7.49	5.96	9.68	1.32	0.00
	9	HCMS		24.1	66.0	98.0	02.52 W	42.13	16.21	17.72	15.16	11.38	6.61
20 mW	10	Frame Rate	00.05 FPS	0.2	✗	✗	15.63 mW	38.51	16.05	26.52	24.72	18.35	7.68
	11	Frame Rate	00.05 FPS	0.2	0.2	✗	16.33 mW	40.94	18.21	32.80	28.39	18.42	8.21
	12	Frame Rate	00.05 FPS	0.2	0.2	0.2	16.02 mW	38.25	16.84	31.24	25.02	16.56	8.10
	13	Greedy		0	0	✗	00.35 mW	23.67	8.29	5.96	9.68	1.32	0.00
	14	Random	τ_{tr}0.99, τ_{inf}1.000, c=0	0	0	0	00.35 mW	20.33	9.12	6.31	6.12	0.00	0.00
	15	AdaMML		0.0 / 100	0.0 / 100	0.0 / 100	2168 mW	23.86	7.84	5.96	9.68	1.32	0.00
	16	HCMS		0	0	98.0	04.92 mW	18.75	10.50	23.73	9.87	2.92	2.12

Sensors: 📹 Video (RGB), 🎙 Audio, 🧭 IMU, 📷 Monochrome, 👁 Gaze.

static baselines; at 20 mW, it is forced to rely exclusively on the IMU and fails to compete with sparsely sampled multimodal cues.

CaptainCook4D: Table 3 evaluates CaptainCook4D [35]. Similar to CMU-MMAC, this scenario is highly susceptible to over-segmentation at high framerates; consequently, the unconstrained 30 FPS baseline (row 1) achieves a low edit score of 18.05 despite consuming 9.30 W. As a result, also in this case, static policies strictly dominate. Under the 2.8 W budget, a minimal 1 FPS RGB-only policy (row 2) achieves best overall performance with an edit score of 36.48 and accuracy of 42.23 using just 0.31 W, outperforming various multimodal fusions at lower framerates (rows 3–5). Dynamic approaches lose their advantage, with Greedy (row 6) and Random (row 7) lagging significantly at 33.80% and 29.74% accuracy, respectively. Complex models like AdaMML fail entirely: the severe tuning required to comply with the budget (2.30 W, row 8) collapses performance to 22.60% accuracy. Meanwhile, HCMS obtains reasonable accuracy but a low edit score, mirroring the over-segmentation issues of the unconstrained baseline.

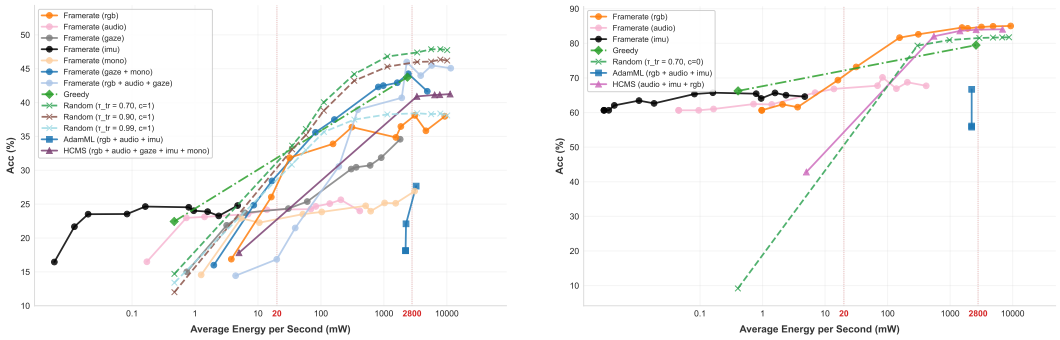
Under the extreme 20 mW constraint, dynamic approaches drop to 0% modality usage to conserve energy, operating blind and plummeting to an edit score of roughly 6.00 (rows 13, 14). In contrast, drastically reduced 0.05 FPS static baselines using RGB alone or fused with audio and IMU (rows 10–12) successfully extract sufficient context to maintain robust edit scores up to 32.80. AdaMML's architectural overhead again proves fundamentally unsuited for ultra-low power applications, consuming over 2.1 W even with 0% sensor usage (row 15). Finally, HCMS adapts to the severe power constraints by relying exclusively on the cheapest available modality.

5.2 Energy-Accuracy trade-offs

Figure 4 illustrates stark domain differences in routing efficiency. On Ego-Exo4D (Fig. 4a), Random routing ($\tau_{tr} = 0.70, c = 1$) strictly dominates, defining the Pareto frontier across the energy spectrum. Static policies exhibit a clear transition: ultra-lightweight IMU performs reasonably at extreme constraints (< 1 mW), while multimodal fusions (e.g., RGB+Audio) overtake them as budgets approach 2.8 W. Conversely, CMU-MMAC (Fig. 4b) inverts this dynamic. Here, a static Framerate (RGB) policy establishes the optimal frontier, rapidly saturating to peak accuracy at minimal energy expenditure. A random policy fails to match this efficiency, collapsing entirely at ultra-low scales. Finally, the trade-off curves expose the severe architectural penalties of learned models. While AdaMML clusters at the extreme right of the energy axis, HCMS is competitive at high energy costs, but collapses when a strict budget is enforced. Sec. A.9 reports trade-off curves for CaptainCook4D.

5.3 Qualitative Examples and Collapse of Learned Policies

Figure 5 illustrates the qualitative limitations of AdaMML [34]. To offset its massive computational overhead, the policy overcompensates by heavily biasing activation toward the cheapest sensors. As shown on the left, the system frequently relies exclusively on the IMU even when visual context



(a) Ego-Exo4D (b) CMU

Figure 4: Energy-Accuracy trade-offs on Ego-Exo4D (a) and CMU (b).

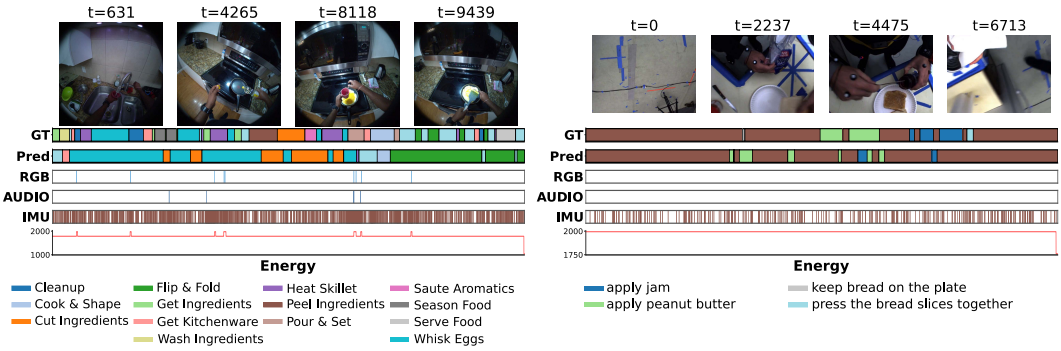


Figure 5: Qualitative examples of success and failure cases for AdaMML [34] in Ego-Exo4D [21] and CMU-MMAC [12] respectively

is strictly required, causing severe segmentation errors. This stems from a structural flaw in the AdaMML loss formulation: it penalizes energy usage during successful predictions but fails to penalize the routing policy when conservative sensor choices cause misclassifications. The right panel shows a complete policy collapse, where the model permanently deactivates RGB and audio, relying solely on the IMU.

6 Conclusions

We introduced Ego-METAS, the first benchmark for online, energy-efficient multimodal temporal action segmentation. By unifying three diverse datasets, we shift the focus of egocentric perception from compute-heavy offline analysis to the energy constraints of real-world wearables. Our evaluations reveal a critical gap: while dynamic modality routing is essential for efficiency, existing policies designed for trimmed clips struggle to adapt to untrimmed, continuous sequences. Ego-METAS provides the foundation to drive the development of adaptive, always-on perception systems.

Limitations: While comprehensive, our energy constraints rely on hardware-profiled estimates rather than physical edge deployment, potentially omitting some complex real-time processing overheads. Additionally, our framework evaluates routing on pre-extracted features; future architectures should jointly optimize the routing policy and feature extractors to maximize true on-device efficiency.

Broader Impact: Energy-efficient perception enables always-on wearable assistive technologies without severe battery drain while simultaneously reducing the carbon footprint of continuous AI. However, always-on cameras inherently raise bystander privacy concerns, dictating that real-world deployment must be strictly coupled with robust on-device privacy-preserving mechanisms.

7 Acknowledgments

This work was supported by projects PID2024-158322OB-I00 , (MCIN/AEI/10.13039/501100011033/ FEDER, UE), project JIUZ2024-IyA-07 and Aragon Government DGA T45-23R.

References

[1] Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3):1–18, 2017.

[2] Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xlstm: Extended long short-term memory. *Advances in Neural Information Processing Systems*, 37:107547–107603, 2024.

[3] Pietro Bonazzi, Sizhen Bian, Giovanni Lippolis, Yawei Li, Sadique Sheik, and Michele Magno. Retina: Low-power eye tracking with event camera and spiking hardware. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5684–5692, 2024.

[4] Junwen Chen, Gaurav Mittal, Ye Yu, Yu Kong, and Mei Chen. Gatehub: Gated history unit with background suppression for online action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19925–19934, June 2022.

[5] Lu Chen, Gun Li, Weisi Xie, Jie Tan, Yang Li, Junfeng Pu, Lizhu Chen, Decheng Gan, and Weimin Shi. A survey of computer vision detection, visual slam algorithms, and their applications in energy-efficient autonomous systems. *Energies*, 17(20):5177, 2024.

[6] Sanjoy Chowdhury, Subrata Biswas, Sayan Nag, Tushar Nagarajan, Calvin Murdock, Ishwarya Ananthabhotla, Yijun Qian, Vamsi Krishna Ithapu, Dinesh Manocha, and Ruohan Gao. Egoadapt: Adaptive multisensory distillation and policy learning for efficient egocentric perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10741–10752, 2025.

[7] Nicola Cottini, Leonardo Gasparini, Marco De Nicola, Nicola Massari, and Massimo Gottardi. A cmos ultra-low power vision sensor with image compression and embedded event-driven energy-management. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 1(3):299–307, 2011.

[8] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Training deep neural networks with low precision multiplications. *arXiv preprint arXiv:1412.7024*, 2014.

[9] Jinyi Cui and Tianyue Zheng. EM^2: Efficient multimodal sensing via adaptive sensor-computation activation. *IEEE Transactions on Mobile Computing*, 2025.

[10] Arnav M Das, Chi Ian Tang, Fahim Kawsar, and Mohammad Malekzadeh. Primus: Pretraining imu encoders with multimodal self-supervision. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

[11] Roeland De Geest, Efstratios Gavves, Amir Ghodrati, Zhenyang Li, Cees Snoek, and Tinne Tuytelaars. Online action detection. In *European Conference on Computer Vision*, pages 269–284. Springer, 2016.

[12] Fernando De la Torre, Jessica Hodgins, Adam Bargteil, Xavier Martin, Justin Macey, Alex Collado, and Pep Beltran. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. 2009.

[13] Guodong Ding, Fadime Sener, and Angela Yao. Temporal Action Segmentation: An Analysis of Modern Techniques. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(2):1011–1030, February 2024. ISSN 0162-8828. doi: 10.1109/TPAMI.2023.3327284. URL `https://doi.org/10.1109/TPAMI.2023.3327284`.

[14] Ruizhou Ding, Zeye Liu, Rongye Shi, Diana Marculescu, and RD Blanton. Lightnn: Filling the gap between conventional deep neural networks and binarized networks. In *Proceedings of the Great Lakes Symposium on VLSI 2017*, pages 35–40, 2017.

[15] Mouser Electronics. Ultra-low-power accelerometer STMicroelectronics MIS2DU12. `https://www.mouser.es/new/semiconductors/sensor-ics/stmicroelectronics-mis2du12-accelerometer/n-6gixyZ2kgkdg`, 2024. Acess: 4th of May of 2026.

[16] 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.

[17] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 203–213, 2020.

[18] Amir Ghodrati, Babak Ehteshami Bejnordi, and Amirhossein Habibian. Frameexit: Conditional early exiting for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15608–15618, 2021.

[19] Abhinav Goel, Caleb Tung, Yung-Hsiang Lu, and George K Thiruvathukal. A survey of methods for low-power deep learning and computer vision. In *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, pages 1–6. IEEE, 2020.

[20] Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass. Ssast: Self-supervised audio spectrogram transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10699–10709, 2022.

[21] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024.

[22] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[23] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[24] Kai Hu, Feng Gao, Xiaohan Nie, Peng Zhou, Son Tran, Tal Neiman, Lingyun Wang, Mubarak Shah, Raffay Hamid, Bing Yin, et al. M-llm based video frame selection for efficient video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13702–13712, 2025.

[25] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[26] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[27] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Thop: Pytorch-opcounter, 2026. URL `https://github.com/ultralytics/thop`.

[28] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6232–6242, 2019.

[29] Hong Li, Xingyu Li, Pengbo Hu, Yinuo Lei, Chunxiao Li, and Yi Zhou. Boosting multi-modal model performance with adaptive gradient modulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22214–22224, 2023.

[30] Zijia Lu and Ehsan Elhamifar. Fact: Frame-action cross-attention temporal modeling for efficient action segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18175–18185, 2024.

[31] Zdravko Marinov, Alina Roitberg, David Schneider, and Rainer Stiefelhagen. Modselect: Automatic modality selection for synthetic-to-real domain generalization. In *European Conference on Computer Vision*, pages 326–346. Springer, 2022.

[32] Meta. Battery life on ai glasses. `https://www.ray-ban.com/usa/l/discover-ray-ban-meta-ai-glasses`. Accessed: 2026-05-05.

[33] Katsuyuki Nakamura, Hiroki Ohashi, and Mitsuhiro Okada. Sensor-augmented egocentric-video captioning with dynamic modal attention. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4220–4229, 2021.

[34] Rameswar Panda, Chun-Fu Richard Chen, Quanfu Fan, Ximeng Sun, Kate Saenko, Aude Oliva, and Rogerio Feris. Adamml: Adaptive multi-modal learning for efficient video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7576–7585, 2021.

[35] Rohith Peddi, Shivvrat Arya, Bharath Challa, Likhitha Pallapothula, Akshay Vyas, Bhavya Gouripeddi, Qifan Zhang, Jikai Wang, Vasundhara Komaragiri, Eric Ragan, et al. Captaincook4d: A dataset for understanding errors in procedural activities. *Advances in Neural Information Processing Systems*, 37: 135626–135679, 2024.

[36] Chiara Plizzari, Gabriele Goletto, Antonino Furnari, Siddhant Bansal, Francesco Ragusa, Giovanni Maria Farinella, Dima Damen, and Tatiana Tommasi. An Outlook into the Future of Egocentric Vision. *International Journal of Computer Vision*, 132(11):4880–4936, November 2024. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-024-02095-7. URL `https://link.springer.com/10.1007/s11263-024-02095-7`.

[37] Shraman Pramanick, Yale Song, Sayan Nag, Kevin Qinghong Lin, Hardik Shah, Mike Zheng Shou, Rama Chellappa, and Pengchuan Zhang. Egovlpv2: Egocentric video-language pre-training with fusion in the backbone. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5285–5297, 2023.

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[39] Laura Romeo, Roberto Marani, Anna Gina Perri, and Juergen Gall. Multi-modal temporal action segmentation for manufacturing scenarios. *Engineering Applications of Artificial Intelligence*, 148:110320, May 2025. ISSN 09521976. doi: 10.1016/j.engappai.2025.110320. URL `https://linkinghub.elsevier.com/retrieve/pii/S0952197625003203`.

[40] Maria Santos-Villafranca, Dustin Carrión-Ojeda, Alejandro Perez-Yus, Jesus Bermudez-Cameo, Jose J. Guerrero, and Simone Schaub-Meyer. Multimodal knowledge distillation for egocentric action recognition robust to missing modalities. In *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, 2026.

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

[42] Yuhan Shen and Ehsan Elhamifar. Progress-aware online action segmentation for egocentric procedural task videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18186–18197, 2024.

[43] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[44] Dipika Singhania, Rahul Rahaman, and Angela Yao. C2f-tcn: A framework for semi-and fully-supervised temporal action segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10): 11484–11501, 2023.

[45] Yu-Chuan Su and Kristen Grauman. Leaving some stones unturned: dynamic feature prioritization for activity detection in streaming video. In *European Conference on Computer Vision*, pages 783–800. Springer, 2016.

[46] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2820–2828, 2019.

[47] Shuhan Tan, Tushar Nagarajan, and Kristen Grauman. Egodistill: Egocentric head motion distillation for efficient video understanding. *Advances in Neural Information Processing Systems*, 36:33485–33498, 2023.

[48] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[49] Naigang Wang, Jungwook Choi, Daniel Brand, Chia-Yu Chen, and Kailash Gopalakrishnan. Training deep neural networks with 8-bit floating point numbers. *Advances in neural information processing systems*, 31, 2018.

[50] Yulin Wang, Yizeng Han, Chaofei Wang, Shiji Song, Qi Tian, and Gao Huang. Computation-efficient deep learning for computer vision: A survey. *Cybernetics and intelligence*, 2024.

[51] Zejia Weng, Zuxuan Wu, Hengduo Li, Jingjing Chen, and Yu-Gang Jiang. HCMS: Hierarchical and Conditional Modality Selection for Efficient Video Recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 20(2):1–18, February 2024. ISSN 1551-6857, 1551-6865. doi: 10.1145/3572776. URL `https://dl.acm.org/doi/10.1145/3572776`.

[52] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*, 2024.

[53] Zihui Xue and Radu Marculescu. Dynamic multimodal fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2575–2584, 2023.

[54] Mingyu Yang, Yu Chen, and Hun-Seok Kim. Efficient deep visual and inertial odometry with adaptive visual modality selection. In *European conference on computer vision*, pages 233–250. Springer, 2022.

[55] Ruichi Yu, Ang Li, Chun-Fu Chen, Jui-Hsin Lai, Vlad I Morariu, Xintong Han, Mingfei Gao, Ching-Yung Lin, and Larry S Davis. Nisp: Pruning networks using neuron importance score propagation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9194–9203, 2018.

[56] Qing Zhong, Guodong Ding, and Angela Yao. Onlinetas: An online baseline for temporal action segmentation. *Advances in Neural Information Processing Systems*, 37:58984–59005, 2024.

[57] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

A Supplementary material

A.1 Motivation for the new benchmark

The proposed benchmark builds upon the initial Energy-efficient multimodal keystep recognition benchmark introduced in Ego-Exo4D [21]. The lack of a publicly released implementation challenges reproducibility and comparative evaluation, and the provided annotations present several important limitations for online Temporal Action Segmentation (TAS). In particular, the initial benchmark utilizes a trimmed subset of annotations extracted from the Keystep benchmark, excluding a substantial number of action segments, particularly those belonging to background class. While this design choice may be reasonable for keystep prediction in pre-segmented clips, in METAS we aim to evaluate continuous perception in long clips, requiring models to handle long background transitions between keysteps. For long procedural videos in the wild, it is crucial to learn to distinguish between true background and foreground, which we found a profound limitation of the existing annotations. Additionally, we observed that the original dataset has some inconsistencies in the annotations, including duplicated classes (see Fig. 7). Taking all these into account, motivated us to perform the re-annotation of the dataset, which is described in Sec. A.2.

We also found that the main evaluation metric chosen for the task in the previous benchmark, the *mean calibrated Average Precision (mcAP)*, proposed initially by De Geest et al. [11], while commonly used in online action detection, is suboptimal for our online setting. Let us analyze Eq. 2:

$$\text{cAP} = \frac{1}{P} \sum_{k} \frac{w \cdot TP(k)}{w \cdot TP(k) + FP(k)} I(k), \quad \text{with } w = \frac{\text{\# negative frames}}{\text{\# positive frames}} \tag{2}$$

where $TP(k)$ and $FP(k)$ are the cumulative true and false positives at rank k, $I(k)$ is an indicator function equal to 1 if the k-th prediction is a true positive and 0 otherwise, and P is the total number of positive frames in the dataset. Notice that it can be rewritten as:

$$\text{cAP} = \frac{1}{P} \sum_{k} \frac{TP(k)}{TP(k) + \frac{FP(k)}{w}} I(k). \tag{3}$$

As the proportion of background frames (negative frames) increases, w grows accordingly, causing the term $\frac{FP}{w}$ to approach zero. Consequently, the reported performance in datasets with lots of background, such as Ego-Exo4D, can be artificially inflated, potentially approaching 100%, despite limited true recognition capability. When training on the full dataset, models tend to collapse to predict the background class, as this circumstance produces deceptively high metric scores while failing to capture relevant actions. This issue becomes more pronounced when training across all scenarios. As shown in Fig. 6, the background class dominates the data distribution, clearly separating itself from the remaining classes. As a result, the metric is biased toward background predictions, masking the model's actual ability to recognize meaningful actions.

To address this issue, our benchmark uses other metrics as primary metrics: Frame Accuracy (Acc), segmental Edit Distance (Edit), and segmental F1 scores. Besides, we make publicly available all our code, baselines and new multimodal annotations for three datasets, including the re-annotated Ego-Exo4D [21].

A.2 Re-annotation pipeline

A.2.1 Ego-Exo4D

Despite the limitations discussed in Sec. A.1, Ego-Exo4D remains the largest egocentric dataset with procedural annotations and multiple modalities. It is also the first to introduce the challenging task of Energy-efficient online multimodal keystep recognition. Building upon this benchmark, we conducted a rigurous revision of the dataset and developed a re-annotation pipeline to adapt it for our requirements.

We first performed a detailed analysis of the structure of the dataset, including number of classes, distribution, frequency of appearance, and length. We covered both the original annotations and the subset of classes for the most similar benchmark, the Keystep recognition benchmark. Following

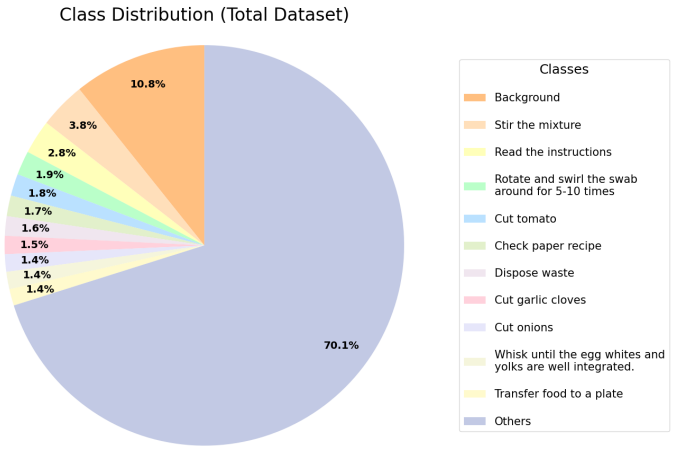


Figure 6: Original Dataset Labels

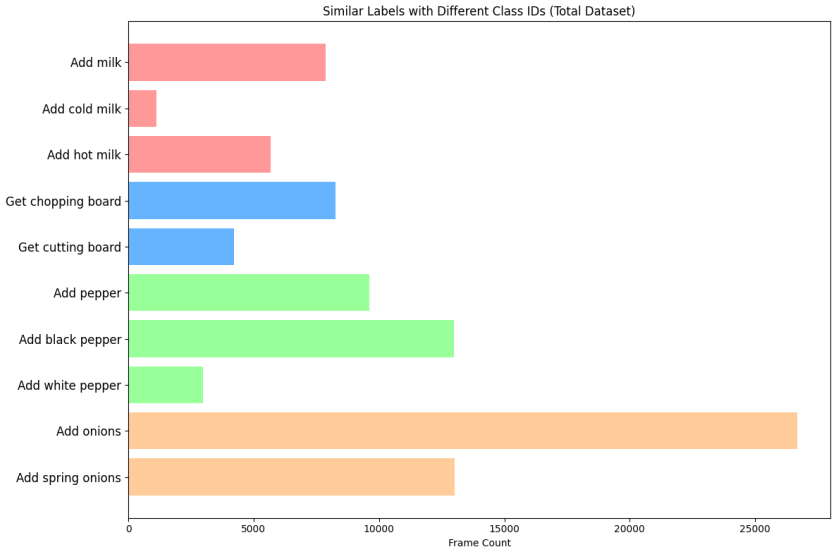


Figure 7: Examples of duplicated classes

prior works [41], we determined the most appropriate strategy for this benchmark was to train a model per scenario and average the results to obtain the global dataset metrics. This methodology prevents the background class to dominate the dataset, discouraging models to trivial collapse.

Even opting for this training strategy, the class granularity was too fine for the limited number of videos per scenario. Besides, many classes were too semantically similar to each other, and background was still the dominant class most of the time, effectively making this task extremely difficult to tackle. In the original annotations, only scenario had enough balance between number of classes and data samples ("First Aid -CPR"). Furthermore, due too lack of sufficient data, we excluded scenarios "Cooking Pasta" and "Cooking Sushi Rolls" since the former is only composed by 3 videos in the training set and the latter is composed of only one video and is impossible to split between training, validation and test.

After identifying the problems, we proceed with class clustering, in order to group similar classes together into a single class. We extracted statistics per class and fed them into an LLM [48] to perform the clustering using this statistical report, the frequency of appearance, and the coherence. This was repeated per scenario, and subsequently human supervision was needed for final corrections and verifications. Final reannotations can be found from Tab. 5 to Tab. 18.

Furthermore, our analysis revealed that, for some videos, the actors were narrating in different languages the steps of the procedure, and other videos have loud music in the background preventing the audio from being informative. Although initial results did not show significant impact in the results, we have documented the names of the compromised videos where this happens (see Tab. 4). To ensure the narrations do not help models to actually predict the correct steps, we made sure that none of this spotted videos are on the test set by swapping the compromised ones with another uncompromised from another split. In addition, we enforce that the training set has all available classes in that split.

Table 4: Actor narration per scenario

Scenario	Found	Total	Percentage (%)
First Aid - CPR	26	51	50.98
Clean and Lubricate the Chain	6	26	23.08
Install a Wheel	9	62	14.52
Remove a Wheel	12	64	18.75
Fix a Flat Tire	24	79	30.38
Covid-19 Rapid Antigen Test	29	182	15.93
Cooking Noodles	7	34	20.59
Making Coffee latte	2	14	14.29
Making Cucumber & Tomato Salad	14	55	25.45
Cooking Scrambled Eggs	7	25	28.00
Cooking an Omelet	9	54	16.67
Making Milk Tea	16	45	35.56
Making Sesame-Ginger Asian Salad	7	29	24.14
Cooking Tomato & Eggs	5	44	11.36
Making Chai Tea	2	9	22.22

Since the test set annotations from Ego-Exo4D were never publicly released, we decided to use the validation set from their benchmark as our test set and split in a 80-20% the training set to obtain a validation set (see Tab. 20). For more information on how to process each modality, please refer to Sec. A.7.

A.2.2 CaptainCook4D

Since we propose a new multimodal egocentric action segmentation benchmark, for CaptainCook4D [35] we selected only those videos that cointain video, audio and IMU information. Therefore, a considerable amount of videos were discarded, and neither of the official proposed splits were large enough for training and evaluating (see Tab. 42). For that reason, we do not train on a per-scenario basis, and instead the full dataset had to be trained jointly. Thus, we create a new split for this benchmark, where 20% of the data is used for testing and the rest is divided 80-20% for train and validation sets. We ensure that the training split contains all different classes, which was not possible for test and validation.

Similarly to Ego-Exo4D, background class was predominant, representing 18% of the dataset, again causing the training models to collapse. Consequently, a new re-annotation process, following previously defined steps was applied. The new annotations can be found at Tab. 19.

Table 5: Label mapping for "Clean and Lubricate the Chain"

New ID	Old ID	Step Name	Old Step Name
0	616	Background	Background
1	436	Prepare Tools & Setup	Check the brake
	442		Get the brush
	443		Get the degreaser
	444		Get the cleaning tool
	445		Fill the cleaning tool with degreaser
	452		Get a cloth
	455		Clean the brush

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
2	446	Apply Degreaser	Add degreaser to the brush
	456		Add degreaser to the chain
3	447	Move Chain (Backpedal)	Slowly move the chain by backpedaling with hand
4	448	Scrub Chain	Hold the brush to the chain as you backpedal with hand
	450		brush the chain while it remains stationary
5	449	Dry Chain	Use a cloth to dry off any liquid on the chain
6	451	Lubricate Chain	Get the chain lube
	453		Place a cloth underneath the chain to catch lube droppings

Table 6: Label mapping for "Install a Wheel"

New ID	Old ID	Step Name	Old Step Name
0	413	Background	Put the bike on a bike repair stand
	414		Put the bike upside down
	420		Return wrenches
	439		Loosen the axle nuts
	616		Background
	956		Deflate the tire
1	408	Get Tools	Get appropriate wrenches
	409		Get bike tire lever
	419		Get bike air pump
	613		Get thru-axle
2	421	Mount Wheel to Frame	Get bike wheel
	422		Put the wheel fully seated into the fork
	423		Get the wheel
	433		Clip the wheel to the hub and axle
	440		Lift the wheel into the fork
3	425	Tighten Axle Nuts	Tighten both axle nuts using a wrench
4	424	Secure Quick Release	Push the release lever inwards
	426		Push level inward and turn axle CW
5	427	Secure Brakes	Tighten brake cable to rear axle
	428		Tighten the break pads
	431		Connect the brake noodle
	437		Clip the brake to the wheel
6	412	Adjust Derailleur	Pivot the derailleur back
7	429	Check Function	Roll the wheel
	430		Rotate the wheel
	432		Check the both hand brakes
	434		Rotate the pedal
	435		Squeeze the brake
	436		Check the brake
	438		Hold hand brake
	441		Check the wheel

Table 7: Label mapping for "Remove a Wheel"

New ID	Old ID	Step Name	Old Step Name
0	410	Background	Shift the gear to the smallest cog
	413		Put the bike on a bike repair stand
	414		Put the bike upside down
	419		Get bike air pump
	616		Background
1	408	Get Tools	Get appropriate wrenches
2	412	Adjust Derailleur	Pivot the derailleur back
3	411	Open Release	Release caliper rim brake pads
	415		Pull the release lever outwards
	416		Loosen both axle nuts using a wrench
	439		Loosen the axle nuts
	602		Pull lever outward and turn axle CCW

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
4	418	Remove Wheel	Pull out the wheel from the frame
5	417	Deflate Tube	Deflate the tube

Table 8: Label mapping for "Fix a Flat Tire"

New ID	Old ID	Step Name	Old Step Name
0	409	Background	Get bike tire lever
	616		Background
1	379	Remove Valve Hardware	Remove dust cap from the wheel
	380		Remove presta valve from the wheel
	381		Remove stem nut from the wheel
	382		Loosen lock nut at the valve stem
	396		Remove the valve cap
	398		Remove the dust cap
2	383	Deflate Tube	Squeeze out air inside the tube
	417		Deflate the tube
3	384	Detach Tire Bead	Engage tire lever on the rim
	385		Pull back and lift bead out of rim
	386		Use tire lever around the rim
4	387	Remove Old Tube	Push the valve through its hole
	388		Pull the inner tube out
5	389	Inspect Tire	Check for damage/splits in thread
	407		Check for damage/splits in tube
6	391	Prepare New Tube	Get a new inner tube
	399		Unpack the new tube
7	400	Install Inner Tube	Engage the valve stem into the rim
	401		Fit the tube to the wheel
8	402	Mount Tire Bead	Fit the tire to the wheel
9	392	Install Valve Hardware	Fix stem nut to the wheel
	394		Tighten the Dust Cap
	395		Attach dust cap to the valve
	397		Apply the dust cap
10	393	Inflate Tire	Inflate the inner tube a little
	403		Partially inflate to check seating
	406		Fully inflate the tire
11	405	Check Bead Seating	Inspect the bead seat line

Table 9: Label mapping for "Covid-19 Rapid Antigen Test"

New ID	Old ID	Step Name	Old Step Name
0	133	Background	Set a timer
	334		Check the expiration date
	350		Blow nose
	357		Get a timer
	363		Wait for testing incubation time
	365		Fold package box
	371		Stop timer
	373		Wait for 15 minutes
	377		Collect saliva sample
	378		Spit into the testing tube
	615		Mistake
	616		Background
	873		Slowly extract tube from mouth
1	355	Unbox package	Unbox package
2	337	Arrange Materials	Arrange test material
	370		Position tube on the box
3	335	Prepare Tube	Locate test tube
	336		Locate and unwrap test tube cap
	338		Unwrap the testing tube

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	342		Get the testing tube
	345		Carefully open tube seal paper
	364		Open the test tube lid
4	343	Fill Tube	Fill tube with testing solution
	349		Add sealed solution
5	344	Prepare Swab	Locate and unwrap swab
	348		Wrap/Unwrap Antigen Test Pack
	351		Slowly insert swab into nose
6	353	Swab Nose	Slowly extract swab from nostril
	354		Repeat with other nostril
7	352	Swab Nose (Rotate)	Rotate swab 5-10 times
	347		Cover the test tube
	358		Dip swab into the testing tube
8	367	Mix Swab	Remove swab from the tube
	369		Shake the test tube
	372		Check solutions tube
	346		Locate and unwrap test plate
9	356	Prepare Cassette	Unwrap testing plate
	366		Locate and unwrap test strip
	368		Dip test strip into tube
	360		Apply drops to testing plate
10	361	Apply Sample	Set swab into testing plate
	362		Cover the testing plate
11	359	Check Result	Visually inspect testing tube
	374		Check the testing plate
12	339	Dispose Waste	Return items to the box
	375		Dispose waste
	340		Read the instructions
13	341	Read Instructions	Fold the instruction paper
	376		Unfold the instruction paper

Table 10: Label mapping for “Cooking Noodles”

New ID	Old ID	Step Name	Old Step Name
	123		Visually inspect the recipe
0	132	Background	Check paper recipe
	133		Set a timer
	616		Background
	207		Get chopping board
	208		Get chopsticks
	210		Get cutting board
	211		Get fork
	213		Get kitchen tong
	214		Get kitchen towel
	215		Get knife
	217		Get measuring tool
	218		Get napkin
1	221	Get Kitchenware	Get plate
	222		Get pot holder
	223		Get sieve
	224		Get skillet/pan/wok
	225		Get spatula
	226		Get spoon
	229		Get a cup or mug
	357		Get a timer
	493		Get a pot or saucepan
	495		Get an electric kettle
	548		Get a bowl
	140		Get cabbage
2	143	Get Ingredients	Get celeries
	144		Get cheese

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	153		Get curry powder
	160		Get garlic cloves
	161		Get garlic paste
	166		Get honey
	167		Get hot sauce
	168		Get ketchup
	176		Get noodles
	177		Get nutmeg
	178		Get oil
	179		Get olive oil
	181		Get onions
	182		Get oyster sauce
	184		Get peanut butter
	185		Get pepper
	189		Get salt
	190		Get scallion
	191		Get sesame oil
	192		Get sesame seeds
	193		Get soy sauce
	195		Get spring onions
	197		Get tamari
	200		Get toasted sesame oil
	202		Get tomato sauce
	205		Get water
	69		Wash garlic cloves
	72		Wash noodles
	75		Wash scallion
	76		Wash spring onions
	83		Peel garlic cloves
	84		Peel fresh ginger
	86		Peel onions
	87		Peel spring onions
3	93	Prepare Ingredients	Cut cabbage
	95		Cut celeries
	98		Cut chili powder
	102		Cut garlic cloves
	107		Cut mustard leaves
	108		Cut noodles
	110		Cut onions
	111		Cut scallion
	113		Cut spring onions
	529		Cut ginger
4	494	Heat Water	Wait until water boils
	532		Check heat of the pot
	533		Add the noodles to boiling water
5	534	Boil Noodles	Stir noodles in the pot
	535		Taste for doneness
6	536	Drain Noodles	Drain the noodles into a strainer
	38		Add oil
	124		Turn on the stove
	125		Adjust the stove heat
	126		Turn off the stove
	127		Lift the lid
7	128	Prepare Skillet	Close the lid
	129		Place skillet on the stove
	130		Remove skillet from the stove
	477		Check heat of the skillet
	486		Tilt and rotate the skillet
	518		Add olive oil to a skillet
	22		Add garlic cloves
8	23	Stir-Fry Aromatics	Add garlic paste
	24		Add garlic powder

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	39		Add onions
	44		Add fresh ginger
	46		Add scallion
	50		Add spring onions
	517		Add mustard leaves to skillet
	36		Add noodles
9	512	Stir-Fry Noodles	Stir the mixture
	1748		Add onion powder to skillet
	537		Add mix of ingredients to bowl
10	538	Mix Sauce (Bowl)	Add mustard leaves to a bowl
	539		Add olive oil to a bowl
	540		Add oyster sauce to a bowl
	5		Add black pepper
	6		Add cabbage
	8		Add celeries
	9		Add cheese
	11		Add chili powder
	17		Add curry powder
	18		Add dried herbs
	28		Add honey
	29		Add hot sauce
	30		Add ketchup
	37		Add nutmeg
11	41	Season Food	Add peanut butter
	42		Add pepper
	45		Add salt
	47		Add sesame oil
	48		Add sesame seeds
	49		Add soy sauce
	52		Add tamari
	55		Add tomato sauce
	56		Add turmeric powder
	57		Add vinegar
	58		Add water
	121		Taste the recipe
	523		Add oyster sauce to skillet
12	228	Serve Food	Transfer food to a plate
	0		Wash hands
	1		Wipe hands
	235		Wash bowl
	238		Wash cutting board
	242		Wash knife
13	244	Wash Items	Wash measuring tool
	247		Wash plate
	248		Wash pot or saucepan
	249		Wash sieve
	250		Wash skillet/frying pan/wok
	251		Wash spatula
	252		Wash spoon
	253		Put away bowl
	254		Put away chopping board
	263		Put away knife
	264		Put away measuring tool
	265		Put away napkin
	267		Put away plate
14	269	Put Away Items	Put away pot or saucepan
	271		Put away sieve
	272		Put away skillet/pan/wok
	273		Put away spatula
	274		Put away spoon
	275		Put away timer
	284		Put away cheese

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	296		Put away garlic cloves
	297		Put away garlic paste
	300		Put away honey
	301		Put away hot sauce
	302		Put away ketchup
	306		Put away noodles
	308		Put away oil
	309		Put away olive oil
	310		Put away onions
	311		Put away oyster sauce
	313		Put away pepper
	315		Put away salt
	316		Put away scallion
	317		Put away sesame oil
	318		Put away sesame seeds
	319		Put away soy sauce
	321		Put away spring onions
	323		Put away tamari
	326		Put away toasted sesame oil
	328		Put away tomato sauce
	332		Wipe kitchen countertop
	333		Throw out trash or waste

Table 11: Label mapping for "Making Coffee latte"

New ID	Old ID	Step Name	Old Step Name
0	529	Background	Cut ginger
	616		Background
1	207	Get Kitchenware	Get chopping board
	209		Get coffee filter
	211		Get fork
	214		Get kitchen towel
	215		Get knife
	217		Get measuring tool
	218		Get napkin
	226		Get spoon
	229		Get a cup or mug
	493		Get a pot or saucepan
	495		Get an electric kettle
	548		Get a bowl
	582		Get french press
	589		Get a milk frothing pitcher
2	147	Get Ingredients	Get chocolate
	150		Get coffee beans
	151		Get coffee grounds
	174		Get milk
	196		Get sugar
	205		Get water
	531		Get cinnamon stick
3	124	Operate Heating Appliance	Turn on the stove
	125		Adjust the stove heat
	126		Turn off the stove
	127		Lift the lid
	128		Close the lid
	129		Place skillet on stove
	130		Remove skillet from stove
	526		Turn on electric kettle
	558		Fill electric kettle
	559		Heat the saucepan
	560		Reduce water level
	561		Place water on the stove

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
4	494	Wait for Boiling	Wait until water boils
	597		Wait until milk boils
5	15	Prepare Filter & Grounds	Add coffee grounds
	557		Operate the grinder
	563		Add instant coffee
	568		Open instant coffee jar
	569		Close instant coffee jar
	570		Add freshly ground coffee
	572		Place the coffee filter
	573		Set pour-over device
	574		Place filter bag over bowl
	575		Rinse the filter paper
	576		Measure coffee grounds
	577		Add grounds to filter
	584		Add grounds to french press
6	58	Pour Water for Brewing	Add water
	562		Pour hot water to mug
	579		Pour hot water through filter
	583		Preheat french press
7	591	Process Milk	Submerge steam wand tip
	592		Turn on steam wand
	593		Steam to desired temperature
	594		Remove steam wand
	595		Fill a pot with milk
	596		Stir the milk in the pot
8	564	Pour Milk	Add cold milk
	565		Add hot milk
	590		Pour milk into pitcher
	598		Pour cold milk into mug
	599		Pour frothed milk into mug
9	566	Extract & Mix Coffee	Mix instant coffee in water
	567		Mix instant coffee in milk
	571		Wait for extraction
	580		Strain coffee into a cup
	581		Remove pour-over device
	585		Stir coffee/water mixture
	586		Place lid on french press
	587		Press the plunger
10	588	Pour Coffee to Serve	Pour coffee into a mug
11	12	Add Sweetener & Spices	Add chocolate
	25		Add fresh ginger
	51		Add sugar
	99		Cut chocolate
	121		Taste the recipe
	233		Add cinnamon stick
	600		Stir the coffee
	601		Remove cinnamon
12	237	Wash Items	Wash cup or mug
	242		Wash knife
	244		Wash measuring tool
	245		Wash napkin
	248		Wash pot or saucepan
	252		Wash spoon
13	253	Put Away Items	Put away bowl
	256		Put away coffee filter
	257		Put away cup or mug
	265		Put away napkin
	269		Put away pot or saucepan
	270		Put away pour-over device
	274		Put away spoon
	287		Put away chocolate
	289		Put away cinnamon sticks

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	290		Put away coffee grounds
	305		Put away milk
	322		Put away sugar
	331		Put away water
	1		Wipe hands
14	332	Wipe Clean	Wipe kitchen countertop
	333		Throw out trash

Table 12: Label mapping for "Making Cucumber & Tomato Salad"

New ID	Old ID	Step Name	Old Step Name
0	124	Background	Turn on the stove
	616		Background
	129		Place the skillet/pan/pot on the stove
	207		Get chopping board
	210		Get cutting board
	215		Get knife
	219		Get peeler
1	221	Get Kitchenware	Get plate
	223		Get sieve
	225		Get spatula
	226		Get spoon
	495		Get an electric kettle
	548		Get a bowl
	134		Get agave
	138		Get black pepper
	145		Get cherry tomatoes
	152		Get cucumber
	154		Get dried herbs
	165		Get green chilies
	170		Get lemon juice
	172		Get lime
	173		Get mayonnaise
	174		Get milk
	178		Get oil
	179		Get olive oil
2	180	Get Ingredients	Get onion powder
	181		Get onions
	185		Get pepper
	188		Get rice vinegar
	189		Get salt
	191		Get sesame oil
	193		Get soy sauce
	196		Get sugar
	200		Get toasted sesame oil
	201		Get tomato
	203		Get turmeric powder
	204		Get vinegar
	206		Get white pepper
	64		Wash cherry tomatoes
	66		Wash cucumber
3	73	Wash Vegetables	Wash onions
	78		Wash tomato
	150		Wash lemon
	81		Peel cucumber
4	85	Peel Ingredients	Peel ice
	86		Peel onions
5	97	Cut Tomato	Cut cherry tomatoes
	115		Cut tomato
6	101	Cut Cucumber	Cut cucumber
7	91	Cut Other Ingredients	Cut black pepper

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	103		Cut green chilies
	105		Cut lime
	110		Cut onions
	118		Remove seeds from cucumber
	120		Remove seeds from tomato
	10		Add cherry tomatoes
	16		Add cucumber
8	39	Add Solid Ingredients	Add onions
	54		Add tomato
	537		Add a mix of ingredients to a bowl
	5		Add black pepper
	11		Add chili powder
	18		Add dried herbs
	31		Add lemon juice
	33		Add mayonnaise
	34		Add milk
	38		Add oil
	42		Add pepper
	45		Add salt
	47		Add sesame oil
9	48	Add Dressing & Spices	Add sesame seeds
	49		Add soy sauce
	51		Add sugar
	56		Add turmeric powder
	57		Add vinegar
	59		Add white pepper
	467		Add agave to a mixing bowl
	471		Add lime juice to a mixing bowl
	472		Add onion powder to a mixing bowl
	510		Pour the dressing
	539		Add olive oil to a bowl
	474		Stir the salad mixture
10	475	Mix Salad	Stir the dressing mixture
	476		Toss the salad in a mixing bowl
11	228	Serve Salad	Transfer food to a plate
	235		Wash bowl
	236		Wash chopsticks
	238		Wash cutting board
12	242	Wash Dishes	Wash knife
	247		Wash plate
	250		Wash skillet or frying pan or wok
	252		Wash spoon
	253		Put away bowl
	254		Put away chopping board
	255		Put away chopsticks
	258		Put away cutting board
	263		Put away knife
	266		Put away peeler
	267		Put away plate
	269		Put away pot or saucepan
	272		Put away skillet or frying pan or wok
	273		Put away spatula
	279		Put away black pepper
13	291	Put Away Items	Put away cucumber
	293		Put away dried herbs
	303		Put away lemon juice
	304		Put away mayonnaise
	308		Put away oil
	310		Put away onions
	313		Put away pepper
	315		Put away salt
	319		Put away soy sauce

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	322		Put away sugar
	327		Put away tomato
	329		Put away turmeric powder
	330		Put away vinegar
14	0	Wipe & Clean	Wash hands
	1		Wipe hands
	332		Wipe kitchen countertop
	333		Throw out trash or waste

Table 13: Label mapping for "Cooking Scrambled Eggs"

New ID	Old ID	Step Name	Old Step Name
0	130	Background	Remove the skillet or pan or pot from the stove
	131		Check phone
	616		Background
1	138	Get Items	Get black pepper
	139		Get butter
	144		Get cheese
	145		Get cherry tomatoes
	146		Get chili powder
	153		Get curry powder
	154		Get dried herbs
	156		Get eggs
	174		Get milk
	178		Get oil
	181		Get onions
	185		Get pepper
	189		Get salt
	195		Get spring onions
	201		Get tomato
	207		Get chopping board
	208		Get chopsticks
	211		Get fork
	215		Get knife
	216		Get ladle
	221		Get plate
	224		Get skillet or frying pan or wok
	225		Get spatula
	226		Get spoon
	227		Get whisk
	548		Get a bowl
2	38	Heat Skillet	Add oil
	124		Turn on the stove
	125		Adjust the stove heat
	126		Turn off the stove
	129		Place the skillet or pan or pot on the stove
	477		Check heat of the skillet
	478		Add butter to skillet
	486		Tilt and rotate the skillet to allow oil or butter flow into empty space
3	76	Prepare Ingredients (Cut/Peel)	Wash spring onions
	78		Wash tomato
	86		Peel onions
	87		Peel spring onions
	92		Cut butter
	96		Cut cheese
	97		Cut cherry tomatoes
	110		Cut onions
	113		Cut spring onions
	115		Cut tomato
4	34	Whisk Eggs	Add milk
	479		Crack eggs into a mixing bowl

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	480		Whisk until the egg whites and yolks are well integrated.
5	10	Fry Vegetables	Add cherry tomatoes
	39		Add onions
	50		Add spring onions
	54		Add tomato
	481		Stir the mix of vegetables in the skillet
6	9	Scramble Eggs	Add cheese
	12		Add chocolate
	33		Add eggs
	482		Gently push cooked portions toward the center of the skillet
	485		Cook until the egg is cooked
	487		Pour egg mixture into a skillet
7	5	Season Food	Add black pepper
	11		Add chili powder
	17		Add curry powder
	18		Add dried herbs
	42		Add pepper
	45		Add salt
	48		Add sesame seeds
	614		Add ingredients to egg mixture
1	228	Serve Food	Transfer food to a plate
9	0	Cleanup	Wash hands
	1		Wipe hands
	127		Lift the lid of a skillet or pan or pot
	128		Close the lid of a skillet or pan or pot
	235		Wash bowl
	240		Wash fork
	242		Wash knife
	247		Wash plate
	250		Wash skillet or frying pan or wok
	251		Wash spatula
	252		Wash spoon
	253		Put away bowl
	254		Put away chopping board
	263		Put away knife
	272		Put away skillet or frying pan or wok
	273		Put away spatula
	279		Put away black pepper
	280		Put away butter
	286		Put away chili powder
	292		Put away curry powder
	293		Put away dried herbs
	294		Put away eggs
	305		Put away milk
	308		Put away oil
	315		Put away salt
	332		Wipe kitchen countertop
	333		Throw out trash or waste

Table 14: Label mapping for “Cooking an Omelet”

New ID	Old ID	Step Name	Old Step Name
0	130	Background	Remove skillet/pan/pot from stove
	132		Check paper recipe
	489		Pour cooked egg mixture into a plate
	492		Cook until the omelet is cooked
	616		Background
	2322		Soak up the excess oil from the skillet.
1	207	Get Kitchenware	Get chopping board
	208		Get chopsticks
	210		Get cutting board

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	211		Get fork
	213		Get kitchen tong
	215		Get knife
	216		Get ladle
	217		Get measuring tool
	218		Get napkin
	221		Get plate
	224		Get skillet or frying pan or wok
	225		Get spatula
	226		Get spoon
	229		Get a cup or mug
	548		Get a bowl
	632		Get whisk
2	135	Get Ingredients	Get almonds
	136		Get beef
	138		Get black pepper
	139		Get butter
	144		Get cheese
	146		Get chili powder
	148		Get cilantro
	153		Get curry powder
	154		Get dried herbs
	155		Get dried seaweed
	156		Get eggs
	160		Get garlic cloves
	161		Get garlic paste
	163		Get ginger garlic paste
	164		Get ginger paste
	165		Get green chilies
	168		Get ketchup
	174		Get milk
	178		Get oil
	180		Get onion powder
	181		Get onions
	183		Get parsley
	185		Get pepper
	186		Get radish
	188		Get rice vinegar
	189		Get salt
	190		Get scallion
	193		Get soy sauce
	196		Get sugar
	201		Get tomato
	202		Get tomato sauce
	203		Get turmeric powder
	204		Get vinegar
	206		Get white pepper
3	60	Wash Ingredients	Wash bell peppers
	67		Wash eggs
	70		Wash green chilies
	73		Wash onions
	74		Wash radish
	78		Wash tomato
4	83	Peel Ingredients	Peel garlic cloves
	86		Peel onions
5	89	Cut Ingredients	Cut beef
	90		Cut bell peppers
	92		Cut butter
	102		Cut garlic cloves
	103		Cut green chilies
	110		Cut onions
	111		Cut scallion
	115		Cut tomato

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	119		Remove seeds from green chilies
	529		Cut ginger
6	38	Heat Skillet	Add oil
	124		Turn on the stove
	125		Adjust the stove heat
	126		Turn off the stove
	127		Lift the lid of a skillet/pan/pot
	128		Close the lid of a skillet/pan/pot
	129		Place skillet/pan/pot on the stove
	477		Check heat of the skillet
	478		Add butter to skillet
	486		Tilt and rotate the skillet
7	34	Whisk Eggs	Add milk
	469		Add ginger garlic paste to a bowl
	472		Add onion powder to a mixing bowl
	479		Crack eggs into a mixing bowl
	480		Whisk egg whites and yolks
8	3	Saute Aromatics	Add beef
	9		Add cheese
	13		Add cilantro
	22		Add garlic cloves
	23		Add garlic paste
	26		Add ginger paste
	27		Add green chilies
	30		Add ketchup
	39		Add onions
	40		Add parsley
	43		Add radish
	46		Add scallion
	49		Add soy sauce
	54		Add tomato
	55		Add tomato sauce
	57		Add vinegar
	58		Add water
	488		Stir fry egg mixture
9	483	Pour & Set	Tilt/rotate skillet for uncooked egg
	487		Pour egg mixture into a skillet
10	482	Cook & Shape	Gently push cooked portions to center
11	490	Flip & Fold	Flip the egg to cook evenly
	491		Fold the omelet in half
12	2	Season Food	Add almonds
	5		Add black pepper
	11		Add chili powder
	14		Add cinnamon powder
	17		Add curry powder
	18		Add dried herbs
	37		Add nutmeg
	42		Add pepper
	45		Add salt
	51		Add sugar
	56		Add turmeric powder
	59		Add white pepper
13	228	Serve Food	Transfer food to a plate
14	0	Cleanup	Wash hands
	1		Wipe hands
	235		Wash bowl
	236		Wash chopsticks
	237		Wash cup or mug
	238		Wash cutting board
	242		Wash knife
	244		Wash measuring tool
	247		Wash plate

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	248		Wash pot or saucepan
	250		Wash skillet/frying pan/wok
	251		Wash spatula
	252		Wash spoon
	253		Put away bowl
	254		Put away chopping board
	258		Put away cutting board
	259		Put away fork
	263		Put away knife
	264		Put away measuring tool
	268		Put away pot holder
	273		Put away spatula
	276		Put away almonds
	277		Put away beef
	279		Put away black pepper
	286		Put away chili powder
	292		Put away curry powder
	294		Put away eggs
	297		Put away garlic paste
	299		Put away green chilies
	305		Put away milk
	308		Put away oil
	310		Put away onions
	313		Put away pepper
	315		Put away salt
	320		Put away spinach
	327		Put away tomato
	329		Put away turmeric powder
	332		Wipe kitchen countertop
	333		Throw out trash or waste
15	615	Mistake	Mistake

Table 15: Label mapping for “Making Milk Tea”

New ID	Old ID	Step Name	Old Step Name
	114		Cut tea bag
	121		Taste the recipe
0	122	Background	Measure the temperature
	123		Visually inspect the recipe
	616		Background
	138		Get black pepper
	151		Get coffee grounds
	154		Get dried herbs
	166		Get honey
	174		Get milk
	177		Get nutmeg
	181		Get onions
	190		Get scallion
	196		Get sugar
	198		Get tea bag
	199		Get tea leaves
	205		Get water
	208		Get chopsticks
1	211	Get Items	Get fork
	213		Get kitchen tong
	215		Get knife
	218		Get napkin
	221		Get plate
	222		Get pot holder
	223		Get sieve
	226		Get spoon

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	229		Get a cup or mug
	231		Get a fine-mesh sieve
	493		Get a pot or saucepan
	495		Get an electric kettle
	527		Get ginger
	531		Get cinnamon stick
	548		Get a bowl
	58		Add water
	124		Turn on the stove
	125		Adjust the stove heat
	126		Turn off the stove
2	127	Heat Water	Lift the lid
	128		Close the lid
	129		Place the skillet/pan/pot on stove
	494		Wait until water boils
	558		Fill an electric kettle with water
	15		Add coffee grounds
	18		Add dried herbs
3	53	Brew Tea	Add tea leaves
	496		Add tea bags to hot water
	562		Pour hot water to a cup or mug
4	497	Steep & Simmer	Steep the tea until ready
	502		Simmer the tea over low heat
5	498	Remove Tea	Remove the tea bag
	499		Drain the tea
6	34	Add Milk	Add milk
	500		Pour milk into the tea
	12		Add chocolate
	28		Add honey
7	37	Add Sweetener & Spices	Add nutmeg
	51		Add sugar
	109		Cut nutmeg
	233		Add cinnamon stick
8	234	Stir Drink	Stir the tea
	501		Stir milk with a spoon
9	230	Pour to Serve	Warm up a cup
	232		Pour the tea into the cup
	0		Wash hands
	1		Wipe hands
	77		Wash tea bag
	236		Wash chopsticks
	237		Wash cup or mug
	238		Wash cutting board
	239		Wash electric kettle
	241		Wash kitchen tong
	242		Wash knife
	248		Wash pot or saucepan
	249		Wash sieve
	250		Wash skillet/frying pan/wok
	252		Wash spoon
	255		Put away chopsticks
	257		Put away cup or mug
	260		Put away kettle
10	261	Cleanup	Put away kitchen tong
	263		Put away knife
	269		Put away pot or saucepan
	274		Put away spoon
	290		Put away coffee grounds
	293		Put away dried herbs
	305		Put away milk
	306		Put away noodles
	307		Put away nutmeg

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	310		Put away onions
	316		Put away scallion
	322		Put away sugar
	324		Put away tea bag
	325		Put away tea leaves
	331		Put away water
	332		Wipe kitchen countertop
	333		Throw out trash or waste
11	615	Mistake	Mistake

Table 16: Label mapping for "Making Sesame-Ginger Asian Salad"

New ID	Old ID	Step Name	Old Step Name
	122		Measure the temperature
	131		Check phone
	132		Check paper recipe
0	496	Background	Place the skillet/pan/pot on the stove
	497		Remove the skillet/pan/pot from the stove
	615		Mistake
	616		Background
	135		Get almonds
	137		Get bell peppers
	142		Get carrots
	143		Get celeries
	145		Get cherry tomatoes
	146		Get chili powder
	148		Get cilantro
	149		Get cinnamon powder
	152		Get cucumber
	157		Get feta cheese
	160		Get garlic cloves
	161		Get garlic paste
	162		Get garlic powder
	165		Get green chilies
	166		Get honey
	169		Get lemon
	170		Get lemon juice
	171		Get lettuce
	173		Get mayonnaise
	175		Get mustard
	178		Get oil
	179		Get olive oil
	181		Get onions
	182		Get oyster sauce
	183		Get parsley
	184		Get peanut butter
	185		Get pepper
1	188	Get Items	Get rice vinegar
	189		Get salt
	190		Get scallion
	191		Get sesame oil
	192		Get sesame seeds
	193		Get soy sauce
	194		Get spinach
	195		Get spring onions
	197		Get tamari
	200		Get toasted sesame oil
	201		Get tomato
	206		Get white pepper
	207		Get chopping board
	208		Get chopsticks

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	210		Get cutting board
	211		Get fork
	213		Get kitchen tong
	214		Get kitchen towel
	215		Get knife
	216		Get ladle
	217		Get measuring tool
	218		Get napkin
	219		Get peeler
	220		Get pitcher
	221		Get plate
	226		Get spoon
	229		Get a cup or mug
	527		Get ginger
	548		Get a bowl
	60		Wash bell peppers
	61		Wash cabbage
	62		Wash carrots
	63		Wash celeries
	64		Wash cherry tomatoes
2	65	Wash Ingredients	Wash cilantro
	68		Wash fresh parsley
	69		Wash garlic cloves
	71		Wash lettuce
	76		Wash spring onions
	528		Wash ginger
	79		Peel carrots
	80		Peel coriander leaves
	82		Peel fresh cilantro
3	83	Peel Ingredients	Peel garlic cloves
	84		Peel fresh ginger
	86		Peel onions
	503		Grate carrots
4	504	Grate Ingredients	Grate ginger
	603		Grate garlic
5	90	Cut Bell Peppers	Cut bell peppers
	116		Remove seeds from bell peppers
	88		Cut almonds
	93		Cut cabbage
	94		Cut carrots
	95		Cut celeries
	98		Cut chili powder
	100		Cut cilantro
	101		Cut cucumber
6	102	Cut Other Veg	Cut garlic cloves
	104		Cut lettuce
	110		Cut onions
	111		Cut scallion
	112		Cut spinach
	113		Cut spring onions
	117		Remove seeds from celeries
	529		Cut ginger
	4		Add bell peppers
	7		Add carrots
	8		Add celeries
	10		Add cherry tomatoes
	13		Add cilantro
	16		Add cucumber
	20		Add feta cheese
7	22	Add Solid Ingredients	Add garlic cloves
	25		Add fresh ginger
	27		Add green chilies

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	32		Add lettuce
	39		Add onions
	46		Add scallion
	473		Add spinach to a mixing bowl
	508		Add prepared mix of vegetables to a bowl
	2		Add almonds
	5		Add black pepper
	11		Add chili powder
	14		Add cinnamon powder
	23		Add garlic paste
	24		Add garlic powder
	26		Add ginger paste
	28		Add honey
	29		Add hot sauce
	31		Add lemon juice
	33		Add mayonnaise
	35		Add mustard
	38		Add oil
8	41	Make Dressing	Add peanut butter
	42		Add pepper
	45		Add salt
	47		Add sesame oil
	48		Add sesame seeds
	49		Add soy sauce
	52		Add tamari
	56		Add turmeric powder
	57		Add vinegar
	59		Add white pepper
	468		Add cardamom to a mixing bowl
	469		Add ginger garlic paste to a mixing bowl
	470		Add lemon to a mixing bowl
	539		Add olive oil to a bowl
	540		Add oyster sauce to a bowl
	506		Whisk the mix of ingredients
9	507	Whisk/Shake Dressing	Shake the mix of ingredients
	509		Whisk the dressing before pouring
10	510	Pour Dressing	Pour the dressing
11	511	Toss Salad	Toss the mixture
	512		Stir the mixture
12	121	Serve Salad	Taste the recipe
	228		Transfer food to a plate
	0		Wash hands
	1		Wipe hands
	235		Wash bowl
	238		Wash cutting board
	240		Wash fork
	242		Wash knife
	244		Wash measuring tool
	246		Wash peeler
	247		Wash plate
	252		Wash spoon
	253		Put away bowl
	254		Put away chopping board
	255		Put away chopsticks
	263		Put away knife
	264		Put away measuring tool
	265		Put away napkin
	267		Put away plate
	274		Put away spoon
	278		Put away bell peppers
	280		Put away butter
13	282	Cleanup	Put away carrots

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	283		Put away celeries
	285		Put away cherry tomatoes
	288		Put away cinnamon powder
	296		Put away garlic cloves
	298		Put away fresh ginger
	299		Put away green chilies
	300		Put away honey
	303		Put away lemon juice
	308		Put away oil
	310		Put away onions
	313		Put away pepper
	314		Put away rice vinegar
	315		Put away salt
	317		Put away sesame oil
	318		Put away sesame seeds
	319		Put away soy sauce
	325		Put away tea leaves
	327		Put away tomato
	332		Wipe kitchen countertop
	333		Throw out trash or waste

Table 17: Label mapping for Cooking Tomato & Eggs

New ID	Old ID	Step Name	Old Step Name
0	132	Background	Check paper recipe
	489		Visually inspect the recipe
	615		Mistake
	616		Background
1	207	Get Kitchenware	Get chopping board
	208		Get chopsticks
	210		Get cutting board
	211		Get fork
	214		Get kitchen towel
	215		Get knife
	217		Get measuring tool
	218		Get napkin
	221		Get plate
	224		Get skillet or frying pan or wok
	225		Get spatula
	226		Get spoon
	227		Get whisk
	229		Get a cup or mug
	493		Get a pot or saucepan
	548		Get a bowl
2	138	Get Ingredients	Get black pepper
	139		Get butter
	143		Get celeries
	145		Get cherry tomatoes
	146		Get chili powder
	156		Get eggs
	158		Get flour
	160		Get garlic cloves
	165		Get green chilies
	178		Get oil
	179		Get olive oil
	183		Get parsley
	185		Get pepper
	189		Get salt
	190		Get scallion
	195		Get spring onions
	196		Get sugar

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	201		Get tomato
	203		Get turmeric powder
	205		Get water
	206		Get white pepper
	527		Get ginger
3	63	Wash Ingredients	Wash celeries
	75		Wash scallion
	76		Wash spring onions
	78		Wash tomato
4	83	Peel Ingredients	Peel garlic cloves
	120		Remove seeds from tomato
	513		Cut a cross on the tomato
5	92	Cut Aromatics	Cut butter
	95		Cut celeries
	102		Cut garlic cloves
	103		Cut green chilies
	111		Cut scallion
	113		Cut spring onions
	529		Cut ginger
6	97	Cut Tomato	Cut cherry tomatoes
	115		Cut tomato
7	479	Crack Eggs	Crack eggs into a mixing bowl
8	480	Whisk Eggs	Whisk until egg whites/yolks are integrated
9	38	Heat Skillet	Add oil
	124		Turn on the stove
	125		Adjust the stove heat
	126		Turn off the stove
	127		Lift the lid of a skillet/pan/pot
	128		Close the lid of a skillet/pan/pot
	129		Place the skillet/pan/pot on the stove
	477		Check heat of the skillet
	478		Add butter to skillet
	486		Tilt and rotate the skillet
10	482	Scramble Eggs	Gently push cooked portions toward center
	483		Tilt/rotate skillet for uncooked egg
	485		Cook until the egg is cooked
	487		Pour egg mixture into a skillet
	514		Break scrambled egg into small pieces
11	515	Transfer Eggs (Out)	Transfer scrambled egg
12	8	Add & Mush Tomatoes	Add celeries
	22		Add garlic cloves
	25		Add fresh ginger
	27		Add green chilies
	39		Add onions
	40		Add parsley
	46		Add scallion
	50		Add spring onions
	54		Add tomato
	58		Add water
	516		Add cloves to a skillet
	518		Add olive oil to a skillet
	519		Leave to cook until tomato becomes mushy
	521		Add cooked tomatoes into the skillet
13	512	Stir-Fry Mixture	Stir the mixture
14	19	Combine Eggs	Add eggs
	520		Add the scrambled egg into the skillet
15	5	Season Food	Add black pepper
	11		Add chili powder
	42		Add pepper
	45		Add salt
	51		Add sugar
	56		Add turmeric powder

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	59		Add white pepper
	121		Taste the recipe
	522		Add flour to the skillet
16	228	Serve Food	Transfer food to a plate
	0		Wash hands
	1		Wipe hands
	235		Wash bowl
	236		Wash chopsticks
	237		Wash cup or mug
	238		Wash cutting board
	240		Wash fork
	242		Wash knife
	243		Wash ladle
	247		Wash plate
	250		Wash skillet or frying pan or wok
	251		Wash spatula
	252		Wash spoon
	253		Put away bowl
	254		Put away chopping board
	257		Put away cup or mug
	259		Put away fork
17	262	Cleanup	Put away kitchen towel
	263		Put away knife
	264		Put away measuring tool
	265		Put away napkin
	272		Put away skillet or frying pan or wok
	273		Put away spatula
	274		Put away spoon
	294		Put away eggs
	296		Put away garlic cloves
	308		Put away oil
	312		Put away parsley
	315		Put away salt
	317		Put away sesame oil
	322		Put away sugar
	327		Put away tomato
	331		Put away water
	332		Wipe kitchen countertop
	333		Throw out trash or waste

Table 18: Label mapping for “Making Chai Tea”

New ID	Old ID	Step Name	Old Step Name
	132		Check paper recipe
	133		Set a timer
0	357	Background	Get a timer
	525		Check the heat
	616		Background
	141		Get cardamom
	154		Get dried herbs
	159		Get fresh thyme
	174		Get milk
	196		Get sugar
	198		Get tea bag
	199		Get tea leaves
	212		Get kitchen shears
1	217	Get Items	Get measuring tool
	223		Get sieve
	226		Get spoon
	229		Get a cup or mug
	231		Get a fine-mesh sieve

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	493		Get a pot or saucepan
	527		Get ginger
	531		Get cinnamon stick
	124		Turn on the stove
	125		Adjust the stove heat
2	126	Stove Operations	Turn off the stove
	129		Place the skillet/pan/pot on the stove
	130		Remove the skillet/pan/pot from the stove
	58		Add water
3	494	Add Water & Boil	Wait until water boils
	524		Fill a cup or mug with water
	18		Add dried herbs
	21		Add fresh thyme
4	25	Add Spices	Add fresh ginger
	48		Add sesame seeds
	233		Add cinnamon stick
	530		Add cardamom to tea
5	53	Add Tea Leaves	Add tea leaves
	496		Add tea bags to hot water
6	502	Simmering	Simmer the tea over low heat
	34		Add milk
7	106	Add Milk	Cut milk
	500		Pour milk into the tea
8	51	Add Sugar	Add sugar
9	234	Stir Mixture	Stir the tea
	501		Stir milk with a spoon
10	232	Strain & Serve	Pour the tea into the cup
	237		Wash cup or mug
	244		Wash measuring tool
11	248	Wash Items	Wash pot or saucepan
	249		Wash sieve
	252		Wash spoon
	1		Wipe hands
	264		Put away measuring tool
	269		Put away pot or saucepan
	271		Put away sieve
	274		Put away spoon
	275		Put away timer
	281		Put away cardamom
12	289	Put Away Items	Put away cinnamon sticks
	293		Put away dried herbs
	305		Put away milk
	318		Put away sesame seeds
	322		Put away sugar
	325		Put away tea leaves
	332		Wipe kitchen countertop
	333		Throw out trash or waste

Table 19: Label mapping for CaptainCook4D [35]

New ID	Old ID	Step Name	Old Step Name
0	0	Background	Background
	16		Continue slicing with floss
	20		Slice one tomato
	39		Peel 4 large garlic cloves
	41		Chop tomato roughly
1	42	Cut Ingredients	Mince the garlic
	58		Crack one egg in the bowl
	61		Chop 1/4 tomato
	62		Chop 1 tsp cilantro
	63		Chop 1/4 medium onion

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	64		Peel 2 garlic cloves
	65		Chop 1 green chilli
	66		Mince peeled garlic cloves
	87		Peel the cucumber
	89		Chop or grate the cucumber
	92		Peel 1 garlic clove
	93		Spiralize 1 medium zucchini
	105		Cut English muffin into two pieces
	117		Chop 1 shallot
	120		Peel 2 garlic cloves
	122		Mince garlic cloves
	123		Slice the mushrooms
	139		Chop 1 strawberry
	149		Cut tomato into two pieces
	150		Chop 2 tbsp cilantro
	152		Crack one egg in a bowl
	169		Peel 2 cloves of garlic
	171		Mince garlic
	174		Slice mushrooms
	178		Slice 1/3 of the bell pepper
	188		Cut tofu into large cubes
	225		Use scissors to cut corner
	227		Chop 1/4 red bell pepper
	245		Chop 1 scallion
	247		Cut avocado into thin slices
	255		Slice two 1/2 inch rounds
	259		Cut 1/4 cup cherry tomatoes
	266		Peel one medium onion
	268		Cut onion into two pieces
	269		Peel 1 garlic clove
	270		Slice 1/8 medium onion
	271		Cut 1/8 garlic clove
	273		Mince 1/8 garlic clove
	295		Peel 1 garlic clove
	296		Peel 1 medium onion
	297		Chop 1 garlic clove
	298		Slice 1/4 medium onion
	311		Prepare the filter insert
	313		Grind the coffee beans
2	29	Add Ingredients	Add corn into bowl
	31		Add 1 tsp pepper powder
	32		Add 1 tsp softened butter
	36		Add 1 tsp salt to the bowl
	37		Add lime juice to the bowl
	45		Add 1/2 tsp cumin seeds
	46		Add 1/4 tsp mustard to the pan
	50		Add 2 tbsp red chili powder
	52		Add tomato puree to the pan
	53		Add 1/2 tsp salt to the pan
	59		Add 1 tbsp milk to the bowl
	60		Add 1/3 tsp salt to the bowl
	69		Add 1/3 tsp salt to the pan
	70		Add chopped onions to the pan
	72		Add chilli to the pan
	73		Add garlic to the pan
	75		Add 1/8 tsp of turmeric
	76		Add tomatoes to the pan
	78		Pour whisked eggs into pan
	82		Add 1 tsp cumin powder
	83		Add 1 tbsp chopped cilantro
	85		Add 1/2 tsp chaat masala
	86		Add 1/4 tsp salt to the bowl
	88		Add 1/4 tsp red chilli powder

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	90		Add cucumber to whisked curd
	96		Add 1 minced garlic clove
	98		Add the zucchini noodles
	99		Add 1/6 cup parmesan cheese
	107		Pour 1 egg into ramekin cup
	127		Add chopped shallot to pan
	129		Add 1/4 tbsp balsamic vinegar
	130		Add 2 cloves minced garlic
	134		Add 1/2 tsp baking powder
	135		Add 1 egg to blender
	136		Add 1 banana to blender
	137		Add 1 heaped tbsp flour
	141		Pour three puddles into pan
	147		Splash maple syrup on plate
	151		Add chopped cilantro to bowl
	153		Add 1/2 tsp black pepper
	159		Pour egg mixture into pan
	164		Add 1/8 tsp black pepper
	165		Add 1/8 cup soy sauce
	167		Add 1/6 cup water to bowl
	168		Add 1 tsp cornstarch to bowl
	170		Add 1/2 tbsp minced ginger
	173		Add 2 cloves minced garlic
	175		Add 1 tbsp honey to bowl
	180		Add sliced mushrooms to skillet
	181		Add broccoli to skillet
	183		Add bell pepper to skillet
	186		Pour the sauce into skillet
	190		Add 1 tbsp olive oil to pan
	191		Add tofu cubes to the pan
	192		Add 1/4 tsp salt to the pan
	207		Measure and add 2 tbsp flour
	208		Add pinch of salt to bowl
	210		Add 1/4 tsp baking powder
	211		Add 1.5 tbsp sugar to bowl
	214		Add 2 tsp vegetable oil
	215		Add 2 tbsp water to bowl
	216		Add 1/4 tsp vanilla extract
	218		Pour batter into mug
	228		Add 1/3 cup cheddar cheese
	230		Add 1 tbsp water to bowl
	234		Add 1/4 tsp pepper to bowl
	235		Add 1/4 tsp salt to bowl
	236		Add 1/2 tbsp butter to bowl
	238		Add 1 tsp Sriracha sauce
	242		Add 1/4 cup mayonnaise to bowl
	243		Add 1 can drained tuna
	246		Add chopped scallion to bowl
	256		Add 1/8 cup mozzarella
	257		Add 1/16 cup basil to bowl
	258		Add 1/4 tsp salt to bowl
	262		Add 1/4 tsp pepper to bowl
	276		Pour sauce over meatballs
	281		Add 1/16 tsp baking soda
	282		Add 4 tbsp flour to mug
	283		Add 1/8 tsp salt to mug
	284		Add 1/8 tsp baking powder
	286		Add 3 tbsp milk to mug
	287		Add 1 tbsp olive oil to mug
	300		Add noodles to the bowl
	304		Add basil to the bowl
	306		Add chopped cilantro to bowl
	320		Pour water into filter

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	322		Pour rest of water over grounds
3	1	Transfer Items	Place tortilla on board
	2		Scoop nut butter from jar
	5		Scoop jelly from the jar
	11		Discard ends of tortilla
	13		Place floss between toothpicks
	17		Place pinwheels on a plate
	40		Take 1 tomato
	56		Take pan off the heat
	57		Transfer to serving bowl
	103		Remove from heat
	115		Place egg over the lettuce
	133		Transfer to serving dish
	145		Transfer to a plate
	146		Serve with strawberries
	148		Take a tomato
	157		Scoop tomatoes from pan
	158		Put tomatoes on serving plate
	162		Transfer omelette to plate
	163		Take 5 broccoli florets
	166		Take 2 cremini mushrooms
	172		Take 1 bell pepper
	195		Remove pan from heat
	199		Briefly remove from heat
	206		Transfer to a serving dish
	209		Place liner inside mug
	222		Carefully remove liner
	229		Place pepper in bowl
	240		Open a can of tuna
	244		Take 1 ripe avocado
	248		Place avocado slices on leaf
	265		Spoon mixture onto bread
	274		Place 5 meatballs on plate
	280		Take a microwavable mug
	289		Take 1 tbsp marinara sauce
	294		Remove noodles from package
	299		Put vegetables in bowl
	310		Place dripper on mug
	312		Place paper filter in dripper
	316		Transfer grounds to filter
	317		Transfer water to a kettle
	324		Discard filter and grounds
4	33	Mix Ingredients	Stir the bowl
	38		Mix contents of bowl well
	43		Puree tomatoes in blender
	51		Mix contents of pan well
	54		Mix tomato puree with pan
	67		Whisk egg mixture in bowl
	79		Mix with spatula for 3 mins
	91		Combine ingredients in bowl
	109		Stir the ramekin cup
	138		Blitz blender for 20 seconds
	154		Beat the contents of bowl
	160		Stir gently with wooden spoon
	176		Whisk contents of bowl
	185		Whisk sauce again
	217		Whisk batter until no lumps
	233		Mix cheese and bell pepper
	237		Mix ingredients of bowl well
	249		Mix contents of bowl
	264		Combine contents of bowl
	267		Mix sauce in small bowl
	278		Stir contents in microwave

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	285		Stir contents in mug well
	307		Mix in flavour packet
	308		Stir noodles until dissolved
	30		Microwave corn for 2 mins
	34		Microwave corn for 3 mins
	44		Heat 3 tbsp oil in pan
	48		Lower the heat
	49		Saute garlic for 2-3 mins
	55		Simmer over low heat
	68		Heat 2 tbsp oil in pan
	71		Saute onions until soft
	77		Cook covered for 1 minute
	94		Heat large pan on medium
	95		Melt 1 tbsp softened butter
	97		Cook garlic until fragrant
	102		Cook for 2 mins for zoodles
	108		Microwave ramekin uncovered
	110		Microwave until egg is set
	124		Heat 1 tbsp oil in skillet
5	140	Cook Food	Melt butter in frying pan
	142		Cook until tops bubble
	144		Cook for 20-30 seconds more
	155		Heat 1 tbsp oil in pan
	156		Cook tomatoes cut-side down
	179		Heat 2 tbsp oil in skillet
	193		Turn on heat to medium
	194		Cook until tofu is browned
	197		Return heat to medium
	198		Cook until tofu turns brown
	202		Return to low heat
	203		Cook pan for 2 minutes
	205		Cook pan until darkened
	219		Microwave batter for 60s
	231		Melt cheese in microwave
	263		Toast bread until charred
	279		Microwave for 1 more minute
	303		Microwave ramen for 4 mins
	27		Measure 2 cups of corn
6	309	Measure Ingredients	Weigh the coffee beans
	314		Measure 12 oz cold water
	4		Clean knife with paper towel
	7		Clean knife with paper towel
	18		Rinse a tomato
	19		Gently dry with towel
7	84	Wash Items	Rinse 1 medium cucumber
	118		Rinse 3 mushrooms
	119		Pat mushrooms dry
	189		Pat tofu dry with towel
	241		Drain excess water from can
	323		Let coffee drain into mug
	22		Garnish with seasoning
	23		Season tomato slices
	24		Season with black pepper
	25		Sprinkle mozzarella on top
	80		Garnish with cilantro
	100		Pepper to taste
	101		Season with salt
	104		Top with more parmesan
	111		Top cup with salsa
8	113	Season Food	Sprinkle cheese on cup
	131		Pepper on pan to taste
	132		Season pan with salt
	200		Drizzle soy sauce on pan

Continued on next page...

New ID	Old ID	Step Name	Old Step Name
	201		Drizzle with sesame oil
	250		Season pepper on bowl
	251		Season bowl with salt
	252		Top lettuce with tuna
	291		Sprinkle cheese on sauce
	292		Sprinkle herbs inside mug
	6		Spread jelly over nut butter
	106		Coat ramekin with spray
	112		Line muffin with lettuce
9	239	Spread Ingredients	Lay out 2 lettuce leaves
	261		Brush bread with olive oil
	290		Spread sauce around surface
	315		Spread open filter to cone
	14		Cross the floss ends
	15		Pull floss ends to slice
	35		Extract lime juice
	121		Pull out mushroom stems
	143		Flip pancakes with spatula
10	196	Manipulate Items	Flip tofu with tongs
	204		Flip tofu on the pan
	220		Invert mug to release cake
	226		Squeeze frosting dollops
	253		Roll up the lettuce wraps
	9		Secure roll with toothpicks
	28		Thaw corn under cold water
	116		Replace top of muffin
	161		Stop stirring to let it set
	177		Set aside sauce mixture
	212		Set aside the lined mug
11	221	Wait	Allow to cool until safe
	254		Secure wrap with toothpick
	301		Cover noodles with water
	302		Cover with lid or towel
	305		Let noodles sit for 1 minute
	321		Wait 30 seconds for bloom

A.3 Dataset Statistics

In Fig. 8 and Fig. 9 detail the total duration (in hours), number of classes, and average video length (in seconds) across all the scenarios for all three datasets after the re-annotation process described in Sec. A.2. In this section, we outline the final details and statistics for the released multi-dataset, comprising the following three collections:

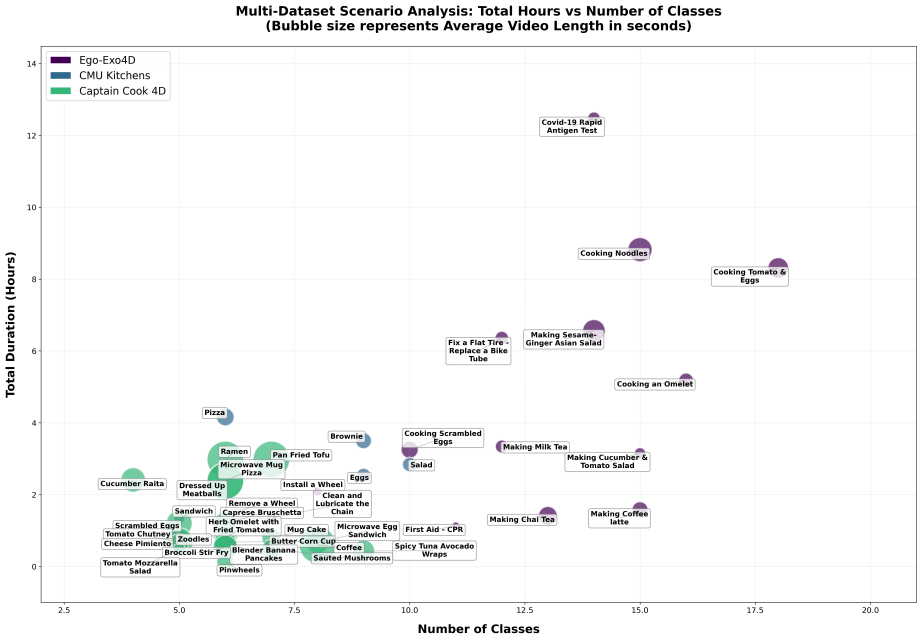


Figure 8: **All Datasets statistics**

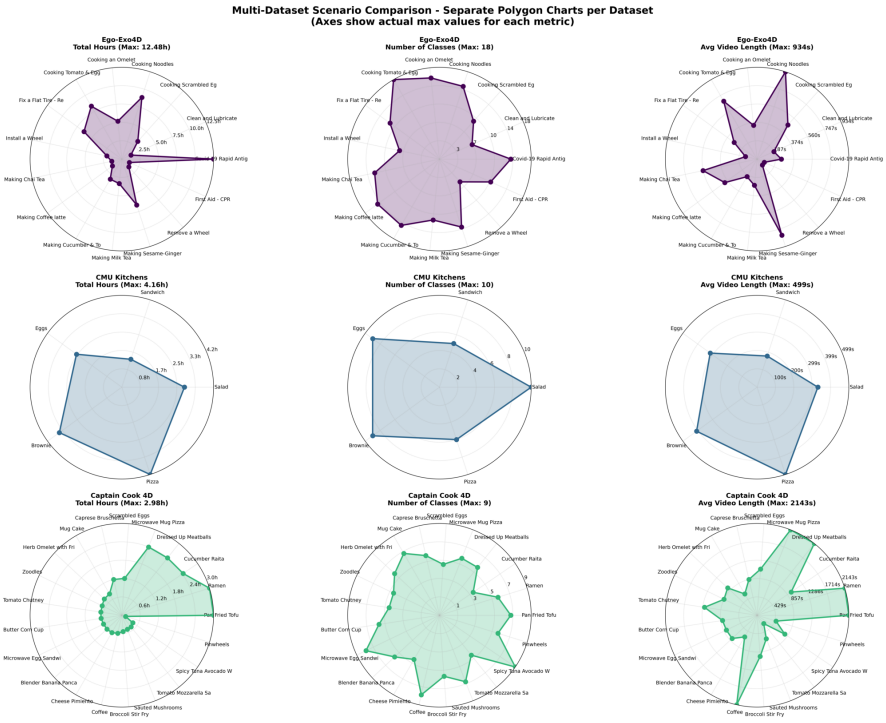


Figure 9: **All Datasets statistics**

Table 20: Ego-Exo4D [21] per scenario statistics

Scenario	N. Videos (Narrations) / duration (min) Train	Val	Test	Classes	Total Time (h)	Avg Len (s)	Min Len (s)	Max Len (s)
Covid-19 Rapid Antigen Test	110 (16)/443.20	27 (13)/113.96	45 (0)/191.56	14	12.48	246.8 ± 83.2	78.6	594.3
Clean and Lubricate the Chain	17 (5)/57.88	4 (1)/9.40	5 (0)/13.77	7	1.35	187.0 ± 73.9	57.8	366.8
Cooking Scrambled Eggs	16 (5)/112.65	3 (2)/31.24	6 (0)/51.36	10	3.25	468.6 ± 244.0	62.6	1000.6
Cooking Noodles	22 (3)/324.94	5 (4)/91.43	7 (0)/112.91	15	8.82	934.0 ± 407.3	333.1	1704.2
Cooking an Omelet	35 (6)/219.40	8 (3)/35.64	11 (0)/55.68	16	5.18	345.2 ± 231.2	109.8	1245.0
Cooking Tomato & Eggs	28 (5)/330.92	7 (0)/72.66	9 (0)/94.97	18	8.31	679.8 ± 411.2	43.1	2079.7
Fix a Flat Tire - Replace a Bike Tube	55 (16)/269.38	13 (8)/64.21	11 (0)/47.91	12	6.36	289.7 ± 137.3	28.3	669.2
Install a Wheel	41 (8)/91.33	10 (1)/15.06	11 (0)/18.30	8	2.08	120.7 ± 103.1	24.3	621.6
Making Chai Tea	6 (2)/57.42	1 (0)/8.84	2 (0)/18.14	13	1.41	562.7 ± 135.4	441.8	922.2
Making Coffee latte	10 (1)/69.46	2 (1)/15.53	2 (0)/9.86	15	1.58	406.5 ± 227.0	29.1	987.9
Making Cucumber & Tomato Salad	34 (8)/109.89	8 (6)/35.35	13 (0)/43.61	15	3.15	206.0 ± 138.4	37.2	509.6
Making Milk Tea	27 (10)/131.28	6 (6)/27.25	12 (0)/41.95	12	3.34	267.3 ± 173.1	74.8	732.5
Making Sesame-Ginger Asian Salad	18 (6)/240.90	4 (1)/50.31	7 (0)/102.49	14	6.56	814.6 ± 340.4	207.1	1383.3
Remove a Wheel	43 (9)/59.14	10 (3)/11.21	11 (0)/14.59	6	1.42	79.6 ± 62.0	19.6	322.6
First Aid - CPR	31 (19)/44.00	7 (7)/9.39	13 (0)/14.57	11	1.13	80.0 ± 23.4	45.5	184.9

Ego-Exo4D [21] This dataset was derived from the energy-efficient set, which we further processed and trimmed as described in Sec. A.2.1. Videos without audio or IMU were removed, resulting in a dataset with 777 total videos. The shortest video has 589 frames (19.6 seconds) and the largest has 62,392 frames (34 minutes, and 39 seconds), yielding a total duration of 66 hours, 24 minutes, and 43.8 seconds. See Tab. 20 for additional per-scenario statistics, including global and independent statistics per split. Regarding number of classes, the dataset has an average of 12.4 classes per scenario. Tabs. 21 to 35 show additional per-class statistics for all scenarios independently.

CMU-MMAC [12]: This dataset was obtained from the official project website. We built upon the framework provided by [30], further partitioning their training set into a 80-20% split to create a validation set. See Tab. 36 for additional per-scenario statistics, including splits. For final data preparation, we isolated the egocentric videos for this study, excluding the third-person perspectives as they contain redundant information and deviate from a true wearable device setting. Since the audio and IMU streams were not originally synchronized with the video and annotations, we aligned them following the methodology of [33]. Videos lacking audio or IMU data were also removed, resulting in a total dataset duration of 14 hours, 22 minutes, and 48 seconds. Regarding video duration, shortest video has 99.1 seconds, whereas longer one has 850.2 seconds, yielding generally shorter video durations compared to Ego-Exo4D. Moreover, average number of classes per scenario is 7.8. See Tabs. 37 to 41 for further per-class statistics for all scenarios independently.

CaptainCook4D [35]: This dataset was taken from the official webpage of [35]. As explained in Sec. A.2.2, videos without Audio or IMU were excluded, resulting in a total of 78 videos and 23 hours and 49 minutes (Tab. 43). The used modalities include video, audio and IMU. Video and audio data were acquired from a GoPro camera system, whereas IMU data were collected with a HoloLens. Due to the synchronization between both devices is not available in the original dataset we had to calibrate this synchronization. After an initial preprocessing step based on visual frame alignment we had to manually refine this synchronization. As shown in Tab. 43), shortest video duration has 247.1 seconds, whereas larger video is 2142.6 seconds long (35 minutes and 43 seconds), similar to Ego-Exo4D. Regarding number of classes, CaptainCook4D has 6.1 unique classes per activity.

Table 21: Per-class stats for "Covid-19 Rapid Antigen Test"

Class ID	Class Name	Instances	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	2680	0.03	163.20	2.22	7.68
1	Unbox package	133	1.47	53.73	15.63	9.58
2	Unbox & Arrange Materials	303	0.70	95.93	11.93	14.57
3	Prepare Tube	310	0.70	60.87	8.23	6.76
4	Fill Tube with Solution	36	10.30	47.10	26.69	8.97
5	Prepare Swab	192	1.60	44.30	12.91	7.55
6	Swab Nose (Insert/Extract)	411	0.07	155.93	5.42	12.52
7	Swab Nose (Rotate)	305	0.47	127.40	15.57	16.61
8	Mix Swab in Tube	431	0.40	60.97	6.35	6.55
9	Prepare Test Cassette	184	1.33	63.10	12.47	7.17
10	Apply Sample to Cassette	186	1.00	67.17	14.84	8.62
11	Check Result	109	1.03	65.57	7.27	9.43
12	Dispose Waste	448	0.83	74.17	10.39	10.19
13	Read Instructions	557	0.03	145.53	12.67	14.38

Table 22: Per-class stats for “Clean and Lubricate the Chain”

Class ID	**Class Name**	**Instances**	**Min (s)**	**Max (s)**	**Avg (s)**	**Std (s)**
0	Background	211	0.03	43.70	3.04	6.50
1	Prepare Tools & Setup	56	0.93	41.77	6.95	8.94
2	Apply Degreaser	45	0.10	45.00	7.55	10.54
3	Move Chain (Backpedal)	56	0.53	109.33	9.87	17.96
4	Scrub Chain	71	1.60	110.90	14.26	17.06
5	Dry Chain	22	4.60	139.97	51.17	42.37
6	Lubricate Chain	35	1.13	110.07	22.89	22.82

Table 23: Per-class stats for “Cooking Scrambled Eggs”

Class ID	**Class Name**	**Instances**	**Min (s)**	**Max (s)**	**Avg (s)**	**Std (s)**
0	Background	694	0.03	20.97	1.55	3.05
1	Get Items	188	0.70	24.20	5.10	4.12
2	Heat Skillet	160	0.47	88.57	10.08	12.22
3	Prepare Ingredients (Cut/Peel)	35	2.83	181.47	36.48	39.41
4	Whisk Eggs	79	1.50	88.83	14.06	11.94
5	Fry Vegetables	50	2.57	61.27	17.01	13.43
6	Scramble Eggs	91	1.83	301.73	35.97	45.65
7	Season Food	41	2.17	48.23	7.74	7.61
8	Serve Food	24	4.67	42.67	19.40	9.25
9	Cleanup	132	0.50	39.73	5.83	6.29

Table 24: Per-class stats for “Cooking Noodles”

Class ID	Class Name	Instances	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	1554	0.03	239.13	3.47	10.16
1	Get Kitchenware	192	0.37	68.70	5.90	7.22
2	Get Ingredients	258	0.30	108.83	6.19	8.72
3	Prepare Ingredients	147	1.63	250.93	39.39	38.48
4	Heat Water	25	2.07	97.47	19.38	22.92
5	Boil Noodles	208	1.00	138.83	15.43	17.22
6	Drain Noodles	38	1.43	53.07	19.56	11.75
7	Prepare Skillet	351	0.07	67.17	6.13	6.76
8	Stir-Fry Aromatics	71	1.60	31.50	10.69	6.00
9	Stir-Fry Noodles	265	0.03	155.13	17.71	19.51
10	Mix Sauce (Bowl)	35	1.03	64.33	16.85	14.08
11	Season Food	203	0.67	72.70	13.98	10.55
12	Serve Food	38	4.23	76.30	21.37	14.42
13	Wash Items	90	0.77	39.50	8.49	7.38
14	Put Away Items	178	0.20	32.13	4.48	4.22

Table 25: Per-class stats for "Cooking an Omelet"

Class ID	Class Name	Instances	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	1376	0.03	49.53	1.38	4.34
1	Get Kitchenware	294	0.37	22.07	4.54	3.17
2	Get Ingredients	312	0.20	34.50	4.05	4.48
3	Wash Ingredients	15	1.80	39.40	9.86	9.66
4	Peel Ingredients	33	1.43	44.93	13.33	10.94
5	Cut Ingredients	116	1.73	115.30	17.21	17.56
6	Heat Skillet	341	0.43	61.20	5.89	6.31
7	Whisk Eggs	173	0.77	55.83	11.54	9.42
8	Saute Aromatics	129	0.83	99.37	12.96	15.88
9	Pour & Set	83	1.53	33.43	10.36	6.54
10	Cook & Shape	50	2.20	105.87	17.82	17.57
11	Flip & Fold	96	0.70	82.07	15.99	15.67
12	Season Food	125	0.87	38.93	6.82	4.27
13	Serve Food	61	1.30	40.23	10.63	8.30
14	Cleanup	213	0.40	112.93	5.03	9.51
15	Mistake	2	7.83	12.33	10.08	2.25

Table 26: Per-class stats for "Cooking Tomato & Eggs"

Class ID	Class Name	Instances	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	1797	0.03	126.73	2.49	6.72
1	Get Kitchenware	266	0.10	23.70	4.44	3.48
2	Get Ingredients	277	0.60	64.20	4.83	5.53
3	Wash Ingredients	24	1.97	33.30	10.24	7.46
4	Peel Ingredients	25	1.57	187.90	32.87	36.94
5	Cut Aromatics	86	2.00	152.07	24.22	22.56
6	Cut Tomato	68	2.00	162.00	34.44	37.32
7	Crack Eggs	47	4.43	39.07	10.98	6.49
8	Whisk Eggs	81	1.80	129.20	18.34	18.92
9	Heat Skillet	459	0.50	57.53	6.02	7.22
10	Scramble Eggs	110	2.63	91.47	19.50	17.86
11	Transfer Eggs (Out)	16	5.60	46.93	18.22	11.88
12	Add & Mush Tomatoes	198	0.97	83.00	10.24	10.80
13	Stir-Fry Mixture	224	0.53	184.23	19.39	20.15
14	Combine Eggs	28	2.63	22.13	9.54	4.98
15	Season Food	97	1.53	36.10	9.18	6.48
16	Serve Food	50	4.70	123.50	20.48	18.24
17	Cleanup	320	0.23	43.03	5.21	5.35

Table 27: Per-class stats for "Fix a Flat Tire - Replace a Bike Tube"

Class ID	Class Name	Instances	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	894	0.03	163.57	2.39	7.15
1	Remove Valve Hardware	86	0.80	26.53	8.09	4.33
2	Deflate Tube	53	1.83	117.47	25.80	18.81
3	Detach Tire Bead	106	0.53	101.03	15.80	16.84
4	Remove Old Tube	96	1.10	80.03	15.59	13.72
5	Inspect Tire	64	1.00	124.63	27.73	27.03
6	Prepare New Tube	88	0.87	81.90	10.76	13.13
7	Install Inner Tube	136	1.97	134.63	28.87	28.89
8	Mount Tire Bead	98	5.60	184.13	30.89	26.28
9	Install Valve Hardware	52	1.33	32.50	8.70	5.53
10	Inflate Tire	166	3.93	139.13	25.38	24.10
11	Check Bead Seating	59	2.23	75.77	19.92	17.89

Table 28: Per-class stats for "Install a Wheel"

Class ID	Class Name	Instances	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	374	0.03	132.97	2.69	8.53
1	Get Tools	37	0.97	89.60	8.17	18.45
2	Mount Wheel to Frame	93	0.90	110.87	14.87	18.62
3	Tighten Axle Nuts	106	1.13	106.73	25.40	22.67
4	Secure Quick Release	32	1.63	38.57	16.49	9.98
5	Secure Brakes	54	2.23	314.40	20.46	44.68
6	Adjust Derailleur	14	2.97	18.63	7.50	4.60
7	Check Function (Spin/Brake)	83	0.53	11.80	4.33	2.52

Table 29: Per-class stats for "Making Chai Tea"

Class ID	Class Name	Instances	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	317	0.03	86.73	3.64	9.60
1	Get Items	81	0.87	32.57	5.45	4.84
2	Stove Operations	70	0.77	28.50	7.04	6.46
3	Add Water & Boil	25	2.57	44.20	10.59	8.17
4	Add Spices	22	1.83	40.97	11.68	9.98
5	Add Tea Leaves	12	2.33	24.07	11.92	6.14
6	Simmering	8	1.57	77.63	29.73	24.83
7	Add Milk	14	3.90	39.00	21.07	11.66
8	Add Sugar	8	8.70	54.00	24.33	13.53
9	Stir Mixture	46	3.43	259.47	21.74	37.68
10	Strain & Serve	9	8.57	39.90	25.93	9.95
11	Wash Items	11	3.50	19.53	10.18	4.22
12	Put Away Items	51	1.20	16.33	4.65	3.03

Table 30: Per-class stats for “Making Coffee latte”

Class ID	Class Name	Instances	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	302	0.03	131.40	4.04	12.75
1	Get Kitchenware	62	0.63	28.57	6.53	5.53
2	Get Ingredients	30	1.17	25.37	5.96	5.13
3	Operate Heating Appliance	53	0.67	31.20	8.68	6.62
4	Wait for Boiling	26	3.50	91.37	26.07	18.25
5	Prepare Filter & Grounds	20	2.73	96.73	19.93	21.45
6	Pour Water for Brewing	25	0.03	48.77	15.13	12.24
7	Process Milk (Heat/Stir/Froth)	35	3.80	64.03	16.76	12.51
8	Pour Milk	20	1.83	96.23	19.66	21.21
9	Extract & Mix Coffee	16	4.73	91.43	20.98	21.86
10	Pour Coffee to Serve	6	8.57	42.37	20.05	12.38
11	Add Sweetener & Spices	26	3.17	20.07	9.92	5.23
12	Wash Items	8	2.20	29.03	11.45	8.89
13	Put Away Items	25	0.60	19.73	4.63	3.66
14	Wipe Clean	9	1.33	29.67	7.85	9.11

Table 31: Per-class stats for “Making Cucumber & Tomato Salad“

Class ID	Class Name	Instances	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	801	0.03	33.27	1.24	3.81
1	Get Kitchenware	121	0.47	27.97	4.43	4.16
2	Get Ingredients	189	0.30	42.80	5.28	7.11
3	Wash Vegetables	39	0.60	34.80	11.71	7.59
4	Peel Ingredients	23	2.53	61.23	18.82	11.70
5	Cut Tomato	97	1.40	89.77	16.62	18.05
6	Cut Cucumber	86	0.03	87.43	24.34	22.57
7	Cut Other Ingredients	25	1.97	61.00	14.55	15.28
8	Add Solid Ingredients	131	0.47	36.67	6.93	7.56
9	Add Dressing & Spices	117	1.53	53.33	8.93	6.69
10	Mix Salad	59	0.87	40.00	14.45	9.75
11	Serve Salad	10	3.13	24.47	11.05	7.49
12	Wash Dishes	22	3.20	65.63	11.68	12.90
13	Put Away Items	72	0.33	30.87	5.07	6.74
14	Wipe & Clean	46	0.40	45.57	6.76	8.35

Table 32: Per-class stats for “Making Milk Tea”

Class ID	Class Name	Instances	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	736	0.03	171.13	3.90	12.12
1	Get Items	249	0.57	104.83	7.19	9.09
2	Heat Water	191	0.20	147.43	9.76	16.30
3	Brew Tea	66	3.60	42.97	12.48	7.83
4	Steep & Simmer	53	1.20	151.03	20.93	25.73
5	Remove Tea	10	1.17	15.67	7.05	3.66
6	Add Milk	48	2.73	22.00	10.34	4.60
7	Add Sweetener & Spices	50	1.47	23.23	9.91	5.37
8	Stir Drink	102	1.67	59.33	9.17	8.32
9	Pour to Serve	29	2.27	41.17	16.66	9.30
10	Cleanup	156	0.57	56.80	7.00	7.72
11	Mistake	1	2.20	2.20	2.20	0.00

Table 33: Per-class stats for "Making Sesame-Ginger Asian Salad"

Class ID	Class Name	Instances	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	964	0.03	88.10	2.83	6.7
1	Get Items	357	0.67	99.83	6.12	8.64
2	Wash Ingredients	55	0.53	29.97	10.19	7.81
3	Peel Ingredients	47	3.40	108.93	28.82	22.15
4	Grate Ingredients	37	7.53	220.37	70.40	47.96
5	Cut Bell Peppers	51	2.23	183.50	30.39	32.99
6	Cut Other Veg	149	2.20	181.20	33.77	30.07
7	Add Solid Ingredients	154	0.90	77.43	12.73	10.90
8	Make Dressing	171	0.07	112.53	14.97	14.05
9	Whisk/Shake Dressing	23	2.17	37.17	16.15	8.65
10	Pour Dressing	22	3.07	50.63	13.22	9.55
11	Toss Salad	50	2.13	61.00	18.88	10.80
12	Serve Salad	16	1.53	44.77	15.28	12.05
13	Cleanup	192	0.30	49.70	6.45	6.66

Table 34: Per-class stats for "Remove a Wheel"

Class ID	Class Name	Instances	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	241	0.03	98.07	6.85	13.00
1	Get Tools	32	0.23	16.87	3.54	3.16
2	Adjust Derailleur	14	3.37	38.57	9.47	8.72
3	Open Quick Release	129	1.60	188.10	22.67	28.01
4	Remove Wheel	24	1.70	21.03	6.83	4.73
5	Deflate Tube	3	14.60	70.73	36.84	24.35

Table 35: Per-class stats for "First Aid - CPR"

Class ID	Class Name	Instances	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	388	0.03	26.07	2.12	4.64
1	Have a conversation asking different qu	24	0.53	16.13	3.34	4.08
2	Check if the patient is responding at a	18	1.17	17.23	4.16	3.58
3	Tap patient to confirm consciousness	75	0.83	12.70	3.82	2.35
4	Confirm patient consciousness from the	98	0.73	13.07	3.82	2.13
5	Kneel on the side of the patient's neck	20	1.27	16.07	4.77	3.30
6	Place the lower part of the palm (heel)	45	0.17	43.47	1.88	6.33
7	Place the other hand on top of the firs	65	0.17	8.97	1.38	1.39
8	Press hard at a rate of 100 to 120 comp	109	0.23	57.40	18.39	8.98
9	Do artificial respirations two times af	4	2.20	6.53	4.84	1.62
10	Call for help	23	1.77	24.93	6.31	5.74

Table 36: CMU Kitchens Dataset Per-Scenario Statistics

	N. Videos / duration (min)							
Scenario	Train	Val	Test	Classes	Total Time (h)	Avg Len (s)	Min Len (s)	Max Len (s)
Salad	19/105.21	6/36.00	6/29.45	10	2.84	330.3 ± 79.8	129.6	527.6
Sandwich	20/60.08	4/10.06	3/9.94	5	1.33	177.9 ± 32.6	99.1	224.4
Eggs	19/97.65	3/16.53	7/38.21	9	2.54	315.3 ± 74.5	179.0	471.6
Brownie	15/95.61	5/37.11	11/77.41	9	3.50	406.7 ± 96.7	213.4	683.7
Pizza	20/167.29	5/34.97	5/47.17	6	4.16	498.9 ± 136.0	242.3	850.2

Table 37: Per-Class Statistics for “Salad”

ID	Class Name	Inst.	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	252	0.07	111.20	27.79	27.14
1	peel carrot	32	2.83	34.43	15.51	7.12
2	cut carrot	33	2.10	113.87	18.65	21.01
3	cut lettuce	20	2.70	42.00	13.44	9.58
4	peel cucumber	32	4.70	40.37	19.52	9.70
5	cut cucumber	35	3.07	27.33	11.29	6.41
6	cut white onion	33	2.87	43.03	17.49	10.29
7	add pepper	15	2.23	10.93	6.30	2.41
8	add mayonnaise	13	3.17	13.17	6.35	2.99
9	mix	8	2.93	22.10	10.20	6.48

Table 38: Per-Class Statistics for “Sandwich”

ID	Class Name	Inst.	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	194	0.10	89.30	18.58	19.30
1	keep bread on plate	26	0.37	5.60	1.45	1.09
2	apply peanut butter	62	1.53	28.00	8.72	6.16
3	apply jam	52	0.70	43.80	10.13	8.94
4	press bread slices	27	1.30	8.17	3.50	1.71

Table 39: Per-Class Statistics for “Eggs”

ID	Class Name	Inst.	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	331	0.23	200.90	20.37	26.22
1	break egg	47	2.67	12.73	5.25	1.72
2	mix	70	1.13	51.57	12.64	10.06
3	add salt	32	1.07	11.87	3.92	2.15
4	add pepper	28	1.50	12.73	4.03	2.40
5	pour oil	31	1.50	10.33	3.98	2.05
6	pour mixture	29	3.57	28.33	10.13	4.82
7	flip	38	1.47	33.47	11.10	6.97
8	plate omelette	29	1.47	13.30	6.61	3.34

Table 40: Per-Class Statistics for “Brownie”

ID	Class Name	Inst.	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	318	0.07	176.43	22.24	24.60
1	break egg	45	3.43	27.83	9.51	5.18
2	mix eggs	14	6.03	32.80	19.56	8.48
3	add water	32	1.90	7.20	3.41	1.17
4	add oil	33	2.07	10.10	4.60	1.91
5	add brownie mix	45	2.47	33.63	13.15	7.68
6	mix contents	67	0.80	133.03	36.96	28.80
7	spray oil	28	3.57	12.13	6.77	2.05
8	pour mixture	39	5.97	66.07	33.70	15.14

Table 41: Per-Class Statistics for “Pizza”

ID	Class Name	Inst.	Min (s)	Max (s)	Avg (s)	Std (s)
0	Background	235	0.10	167.60	29.42	32.53
1	spread batter	40	2.77	128.93	34.05	23.21
2	apply sauce	36	1.73	106.23	38.44	21.64
3	add cheese	50	11.57	214.13	53.76	40.18
4	add pepperoni	49	0.70	228.13	51.61	48.61
5	to oven	30	1.00	8.50	2.97	1.62

Table 42: Captain Cook 4D: Videos per Scenario across Split Strategies

Scenario	Recordings			Person			Environment		
	Train	Val	Test	Train	Val	Test	Train	Val	Test
Blender Banana Pancakes	2	0	1	1	1	1	3	0	0
Broccoli Stir Fry	1	0	1	1	1	0	2	0	0
Butter Corn Cup	2	0	1	1	1	1	3	0	0
Caprese Bruschetta	3	0	2	4	0	1	2	0	3
Cheese Pimiento	3	1	0	2	0	2	2	2	0
Coffee	1	0	0	0	1	0	0	1	0
Cucumber Raita	5	1	3	3	3	3	5	4	0
Dressed Up Meatballs	1	1	2	3	1	0	3	0	1
Herb Omelet with Fried Tomatoes	0	1	2	2	0	1	3	0	0
Microwave Egg Sandwich	2	1	0	1	0	2	2	0	1
Microwave Mug Pizza	3	0	1	1	2	1	3	1	0
Mug Cake	3	0	2	4	0	1	1	0	4
Pan Fried Tofu	3	0	2	2	1	2	2	3	0
Pinwheels	1	0	0	0	1	0	0	1	0
Ramen	3	0	2	3	1	1	4	1	0
Sauted Mushrooms	2	0	1	2	1	0	3	0	0
Scrambled Eggs	1	1	2	2	1	1	1	0	3
Spicy Tuna Avocado Wraps	1	1	0	2	0	0	1	0	1
Tomato Chutney	1	0	1	2	0	0	1	1	0
Tomato Mozzarella Salad	4	2	1	4	0	3	5	2	0
Zoodles	2	0	1	2	0	1	1	0	2
TOTAL	**44**	**9**	**25**	**42**	**15**	**21**	**47**	**16**	**15**

Table 43: Captain Cook 4D Dataset Per-Activity Statistics

	N. Videos / duration (min)							
Activity	Train	Val	Test	Unique classes	Total (h)	Avg Len (s)	Min Len (s)	Max Len (s)
Pan Fried Tofu	3/107.13	0/0.00	2/71.42	7	2.98	2142.6	2142.6	2142.6
Ramen	3/107.13	0/0.00	2/71.42	6	2.98	2142.6	2142.6	2142.6
Cucumber Raita	5/80.17	1/16.03	3/48.10	4	2.41	962.1	962.1	962.1
Dressed Up Meatballs	1/35.71	1/35.71	2/71.42	6	2.38	2142.6	2142.6	2142.6
Microwave Mug Pizza	3/107.13	0/0.00	1/35.71	6	2.38	2142.6	2142.6	2142.6
Scrambled Eggs	1/18.01	1/18.01	2/36.01	5	1.20	1080.4	1080.4	1080.4
Caprese Bruschetta	3/42.89	0/0.00	2/28.59	6	1.19	857.8	857.8	857.8
Mug Cake	3/28.72	0/0.00	2/19.14	7	0.80	574.3	574.3	574.3
Herb Omelet with Fried Tomatoes	0/0.00	1/15.64	2/31.28	6	0.78	938.3	938.3	938.3
Zoodles	2/28.59	0/0.00	1/14.30	5	0.71	857.8	857.8	857.8
Tomato Chutney	1/20.72	0/0.00	1/20.72	5	0.69	1243.3	1243.3	1243.3
Butter Corn Cup	2/27.25	0/0.00	1/13.63	6	0.68	817.6	817.6	817.6
Microwave Egg Sandwich	2/26.57	1/13.28	0/0.00	8	0.66	797.0	797.0	797.0
Blender Banana Pancakes	2/26.57	0/0.00	1/13.28	6	0.66	797.0	797.0	797.0
Cheese Pimiento	3/29.18	1/9.73	0/0.00	5	0.65	583.5	583.5	583.5
Coffee	1/35.71	0/0.00	0/0.00	8	0.60	2142.6	2142.6	2142.6
Broccoli Stir Fry	1/16.03	0/0.00	1/16.03	6	0.53	962.1	962.1	962.1
Sauted Mushrooms	2/19.62	0/0.00	1/9.81	7	0.49	588.7	588.7	588.7
Tomato Mozzarella Salad	4/16.48	2/8.24	1/4.12	5	0.48	247.1	247.1	247.1
Spicy Tuna Avocado Wraps	1/13.03	1/13.03	0/0.00	9	0.43	781.5	781.5	781.5
Pinwheels	1/7.71	0/0.00	0/0.00	6	0.13	462.5	462.5	462.5

A.4 Estimation of Energy Budget per second

For the computation of energy for a video of a given length, we follow Eq. (2).

The constants $\alpha = 4.6 * 10^{-9}$ mJ/MAC and $\beta = 80 * 10^{-9}$ mJ/byte convert these operations to energy (mJ) [21]. We adopt $P_{\text{RGB}} = 15\,\text{mW}$ and $P_{\text{audio}} = 0.5\,\text{mW}$ from [21], and set $P_{\text{mono}} = 1\,\text{mW}$ [7], $P_{\text{IMU}} = 0.2\,\text{mW}$ [15], and $P_{\text{gaze}} = 0.63\,\text{mW}$ [3].

For the memory energy, we use the PyTorch Profiler tool to compute the memory cost. This profiler returns the CUDA bytes used for a forward pass of it through the model. Since this number of bytes is not linear and cannot be estimated, we use this tool per video to obtain the number of bytes, and from it the $E_{\mathcal{V}_{\text{mem}}}$.

The number of Multiply-Accumulate Operations (MACs) is estimated at the beginning of training using the THOP profiler [27]. For feature extractors f_ϕ^{MAC} we use the MACs obtained from a forward pass. For the models used as baselines for this proposed benchmark, we observed that the MAC count scales linearly with the input length, never passing through the origin. Therefore, at the beginning of the training, we pass two inputs of different lengths through the model to estimate the per-frame computational cost and the fixed computational overhead:

$$\mathrm{f}_\Psi^{\text{MAC}}(L) = m_{\text{frame}} L + n_{\text{fixed}},$$

where L denotes the input length, m_{frame} represents the additional computational cost introduced by each extra frame, and n_{fixed} accounts for fixed computations that are independent of the input length. After obtaining the total number of MACs, the computational energy $E_{\mathcal{V}_{\text{comp}}}$ can be obtained.

Ultimately, for the computation of the capture energy $E_{\mathcal{V}_{\text{cap}}}$, no additional functionality was needed, just the number of active steps $a_t^{(m)}$ of each sensor, the temporal duration between consecutive timesteps Δt and its power consumption P_m.

The average energy per second of the video is finally the total energy divided by the number of seconds of the video T.

A.5 Experimental settings

To ensure reproducibility, we provide the specific training configurations used for our baselines in all three datasets. We train per scenario on Ego-Exo4D [21] and CMU-MMAC [12], since joint training causes background class to strongly dominate and can lead to collapse. However, for CaptainCook4D [35], we perform global training across all scenarios simultaneously as there are not enough data to be trained per scenario (details in Sec. A.2). For the energy computation, we follow the formulation and energy values in Sec. 3.2 using the profiler implementation described in Sec. A.4.

All baselines are implemented in PyTorch and trained with the Adam optimizer. We use a weight decay of e^{-4}, an initial learning rate of $5e^{-4}$, and a cosine annealing learning-rate schedule. For non-learned policies, we train for 50 epochs following the training protocol of ProTAS [42]. For learned policies, we adopt the setup of AdaMML [34]: the TAS model is first warmed up for 5 epochs, after which AdaMML alternates policy and task-model training for 20 epochs, followed by 10 epochs of TAS fine-tuning. For HCMS [51], since no specific training pipeline is provided, we replace the alternating scheme with joint training while keeping the remaining AdaMML training specifications unchanged. Experiments were conducted on an NIVIDA RTX 4090 GPU.

For AdaMML [34] and HCMS [51], each trained policy produces a fixed trade-off between performance and energy consumption. Since the target energy budget is not enforced during training within their pipelines, we adapt these policies to different budget constraints at inference time by varying the threshold applied to the Gumbel-Softmax decisions [26]. This allows us to shift the modality-selection behavior of the policy and obtain operating points with different energy-accuracy trade-offs. Similarly, the Random policy was ablated using different thresholds and training and inference time to evaluate their performance under different circumstances.

A.6 Random Policy computation

Random policy ($c = 0$) just performs a random dropout of each modality per frame. However, the cost-aware version ($c = 1$), takes into account the cost of each modality, enforcing more expensive ones to be dropped more times than the cheaper ones while maintaining the dropout percentage τ of the standard ($c = 0$) version.

The percentage of times the average of the modalities should be on is $p_{\text{on}} = 1.0 - \tau$

For each modality $m \in \{1, \ldots, M\}$ with associated cost c_m, we compute log-normalized costs $\log c_m$. And define the cost range as $\Delta \log c = \max_m \log c_m - \min_m \log c_m$

For each modality, we compute relative expensiveness:

$$e_m = \frac{\log c_m - \min \log c}{\Delta \log c} \tag{4}$$

This is transformed into relative selection weights using a weighted interpolation:

$$w_m = \alpha_{\min} + (1 - \alpha_{\min})(1 - e_m) \tag{5}$$

where $\alpha_{\min} = 0.25$ ensures all modalities remain sampleable even for expensive modalities.

The weights are then normalized and, finally, the selection probability for each modality is $\tau_m = p_{\text{on}} \cdot \tilde{w}_m$

A.7 Per Modality Computation

Per each modality, we extract the feature as follows:

Video: We employ DINOv3 [43] as feature extractor, which as default takes 448×448 input images at each timestep t. Feature extraction is applied at each timestep t in the video stream as $\mathbf{x}_t^{(RGB)} = \phi_{RGB}(\mathbf{X}_t^{(RGB)})$. For monochrome cameras we use low-resolution 256×256 images to serve as an energy-efficient alternative modality. We use ViT-B version of DINOv3, which consumes approximately $E_{total}^{(RGB)} = 9.3\,\text{J}$ per complete second of active sensing and computation. The low-resolution alternative reduces this to $E_{total}^{(Mono)} = 3.09\,\text{J}$ per active second.

IMU: Accelerometer and gyroscope raw data from one IMU is directly used as a temporal input signal in PRIMUS [10]. Particularly, since PRIMUS is pre-trained in Ego-Exo4D on 5 second-length temporal signals, for each timestep t, we take a signal of bounds $[t - 5s, t]$. The energy consumption is very low ($E_{total}^{(IMU)} = 4.45\,\text{mJ}$ per active sensor) given the simple architecture and efficient sensing. The IMU configuration (location and placement) varies depending on the acquisition set-up of each dataset. For CMU we use the right wrist IMU, for ego-exo4D we use the embedded left-IMU of the Aria headset and finally for CaptainCook4D we use the embedded IMU of the HoloLens systems.

Audio: We use SSAST [20] audio features as follows: for each timestep t, we take a clip segment corresponding to the last second of audio up to t. This clip is converted into a Mel spectrogram, which is the required input for the SSAST backbone. : $\mathbf{x}_t^{(Audio)} = \phi_{Audio}(\text{spectrogram}(\mathbf{X}_t^{(Audio)}))$. We use the SSAST version where the spectrogram is splitted into a sequence of 16 × 16 patches.The total energy consumption is $E_{total}^{(Audio)} = 414\,\text{mJ}$ per active second. Notice that, while SSAST is typically pre-trained on 10-second clips, we reduce clip length down to just 1 second to reduce compute and memory, and allow the policy to save energy in a more granular way.

Gaze: For gaze modality, instead of using the 2D on the image location as a signal, we cropped a 192x192 bounding box around the 2D gaze position. That allows us to use local visual information which could be relevant for action recognition since this crop represents the region of the image the user is interacting with or the user's interest. This crop is also encoded with Dinov3 but only consuming 1.80 W due to its reduced size.

A.8 Additional insights

The scenarios within each dataset differ substantially, increasing the diversity of our benchmark and introducing a range of distinct challenges. For instance, "Clean and Lubricate the Chain" is the only Ego-Exo4D scenario in which audio hurts performance. While combining modalities often improves performance, noisy or uninformative signals can instead degrade it. This effect is shown in Tab. 44: RGB alone achieves strong performance (rows 1, 7), whereas adding audio substantially reduces performance (rows 2, 3). Although it is true that using a policy that allows to disentangle audio from the rest of the modalities, like Random, allows the model to ignore it signals and get a good performance (rows 13, 15). Policies that activate all modalities at the same timesteps, such as Greedy or Frame Rate, cannot learn to ignore audio independently. Conversely, learned policies such as HCMS and AdaMML are also unsuited to this scenario (rows 4, 6). HCMS requires all lower-cost modalities to be active before enabling RGB, preventing the policy from selecting only the useful modality. AdaMML faces a different limitation: its official code implementation uses a loss that penalizes using too many modality usage only when the prediction is correct. As a result, when predictions are poor, AdaMML tends to activate all modalities and fails to disentangle their individual contributions (see Fig. 10).

A.9 Energy-Accuracy trade-off

Figures 11 12 and 13 visualizes the Energy-Accuracy trade-offs across selected policies for Ego-Exo4D, CMU and CaptainCook4D, illustrating stark differences in routing dynamics across domains. In Ego-Exo4D (Fig. 11), we observe a clear hierarchy where dynamic policies consistently dominate. The Random routing policies ($\tau_{tr} = 0.70$ and 0.90) maintain the highest performance frontier across nearly the entire energy spectrum, demonstrating their robustness for complex, procedural tasks. Static framerate policies exhibit distinct crossing points: ultra-lightweight modalities like IMU provide the best accuracy at extreme energy constraints (< 1 mW),

Table 44: Performance on scenario "Clean and Lubricate the Chain" Ego-Exo4D [21] across energy budgets, policies and parameters.

Budget	#	Policy	Parameters						Energy	Acc	mAP	Edit	F1@10	F1@25	F1@50
No Budget	1	Frame Rate	30.00 FPS	100.0	✗	✗	✗	✗	09.30 W	32.34	32.98	14.57	5.52	4.38	4.00
	2	Frame Rate	30.00 FPS	✗	100.0	✗	✗	✗	00.41 W	0.84	28.93	0.00	0.00	0.00	0.00
	3	Frame Rate	30.00 FPS	100.0	100.0	✗	✗	✗	09.72 W	0.02	28.82	0.00	0.00	0.00	0.00
	4	HCMS		69.1	92.9	100.0	90.3	79.2	13.53 W	0.00	31.95	0.00	0.00	0.00	0.00
	5	Frame Rate	06.00 FPS	20.0	20.0	20.0	20.0	20.0	03.54 W	0.00	25.79	0.00	0.00	0.00	0.00
2.8 W	6	AdaMML		1.4	1.6 / 100	1.3 / 100	100.0	✗	02.34 W	0.00	25.14	0.00	0.00	0.00	0.00
	7	Frame Rate	06.00 FPS	20.0	✗	✗	✗	✗	01.86 W	11.33	27.72	3.70	4.44	4.44	0.00
	8	Frame Rate	10.00 FPS	✗	33.3	✗	✗	✗	00.14 W	0.85	32.86	0.00	0.00	0.00	0.00
	9	Frame Rate	06.00 FPS	20.0	20.0	✗	✗	✗	01.94 W	0.00	28.68	0.00	0.00	0.00	0.00
	10	Frame Rate	06.00 FPS	20.0	20.0	20.0	✗	✗	01.95 W	0.00	31.22	0.00	0.00	0.00	0.00
	11	Frame Rate	06.00 FPS	✗	20.0	20.0	20.0	20.0	01.68 W	0.00	33.65	0.00	0.00	0.00	0.00
	12	Greedy		13.4	13.4	13.4	13.4	13.4	02.36 W	1.54	39.42	7.54	0.00	0.00	0.00
	13	Random	τ_{tr}0.70, τ_{inf}0.900, c=0	10.3	10.1	10.3	10.0	10.0	01.80 W	53.24	35.95	24.16	10.06	7.59	3.14
	14	Random	τ_{tr}0.97, τ_{inf}0.900, c=1	5.2	11.6	20.7	8.7	7.6	01.16 W	46.23	32.75	25.17	12.97	9.17	4.72
20 mW	15	Frame Rate	00.05 FPS	✗	0.2	✗	✗	✗	00.74 mW	1.28	27.53	0.00	0.00	0.00	0.00
	16	Frame Rate	00.05 FPS	0.2	✗	✗	✗	✗	16.52 mW	12.52	28.56	2.03	2.91	2.91	0.00
	17	Random	τ_{tr}0.70, τ_{inf}1.000, c=0	0	0	0	0	0	00.52 mW	57.74	29.25	8.70	14.63	11.43	0.00
	18	Frame Rate	00.05 FPS	0.2	0.2	✗	✗	✗	17.25 mW	0.86	22.68	0.00	0.00	0.00	0.00
	19	Random	τ_{tr}0.70, τ_{inf}1.000, c=1	0	0	0	0	0	00.48 mW	57.74	29.25	8.70	14.63	11.43	0.00

Sensors: Video (RGB), Audio, IMU, Monochrome, Gaze.

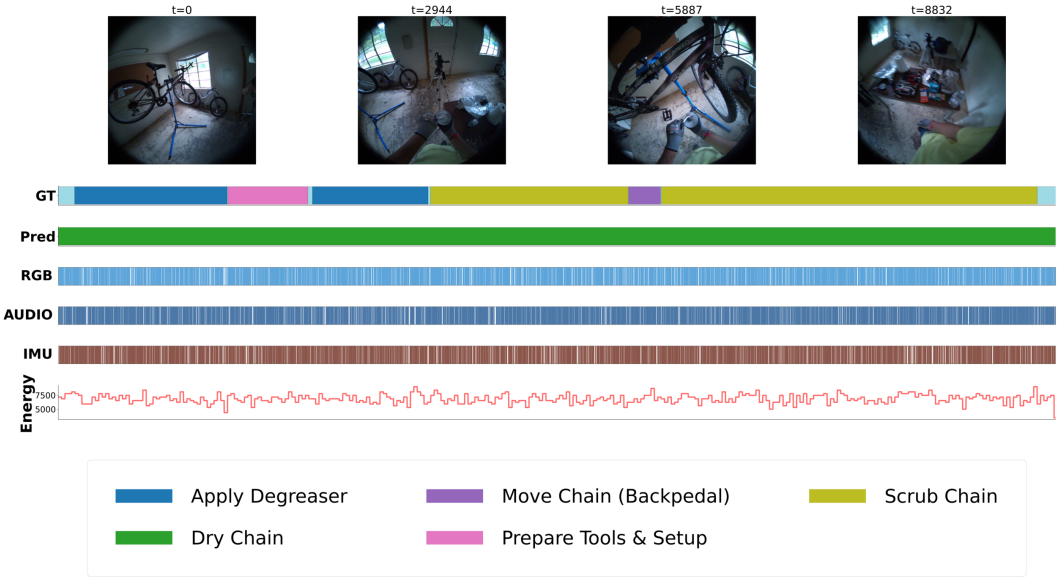


Figure 10: Qualitative example of AdaMML in scenario "Clean and Lubricate the Chain" of Ego-Exo4D [21]

while multimodal fusions (e.g., RGB + Audio or Gaze + Mono) rapidly overtake them as the energy budget relaxes toward 2.8 W.

Conversely, the trends in CMU (Fig. 12) tell a different story. Dynamic approaches like Greedy and Random ($\tau_{tr} = 0.90$) fail to match this static efficiency, collapsing completely at the lowest energy scales. Furthermore, the plot highlights the severe architectural penalty of complex learned models; methods like HCMS, AdaMML, and xLSTM are clustered at the extreme right of the energy axis, fundamentally unable to scale down to the efficiency levels achieved by simpler baselines.

For Captain Cook 4D (Fig. 13) Frame Rate policies (rgb and rgb+audio) clearly overcome dynamic and learned policies across the entire energy spectrum.

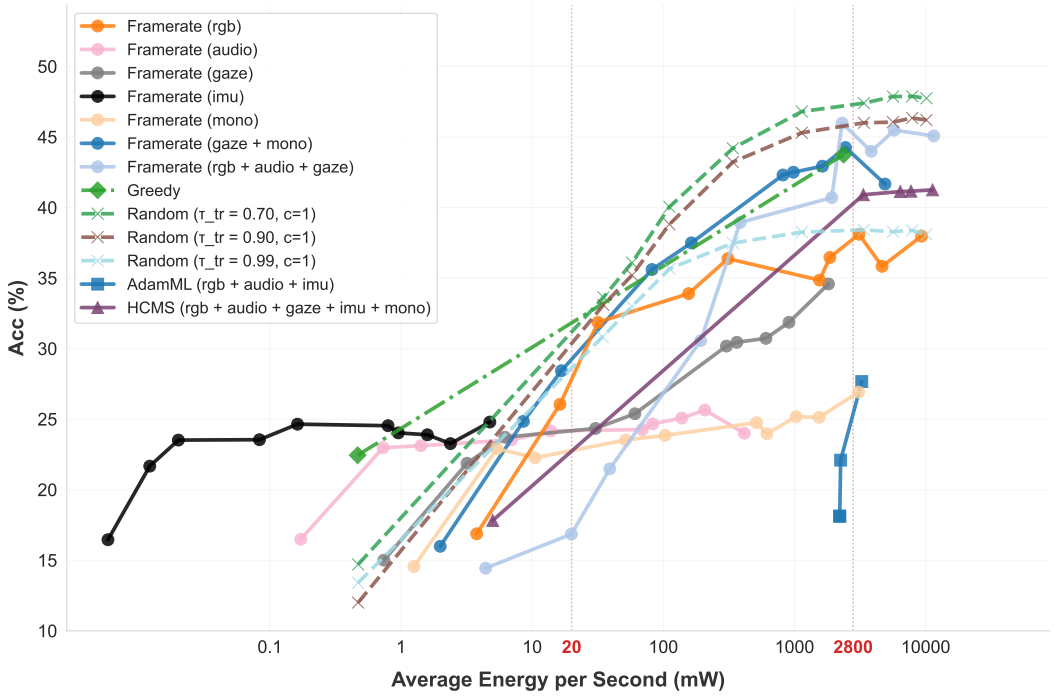


Figure 11: Energy-Accuracy trade-offs on Ego-Exo4D

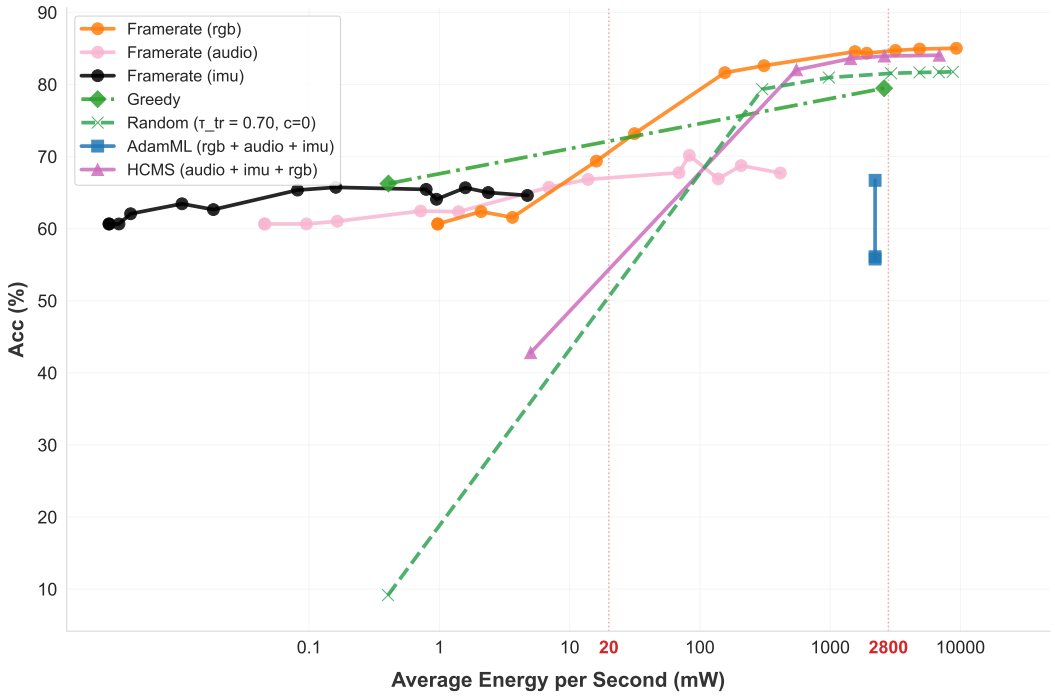


Figure 12: Energy-Accuracy trade-offs on CMU

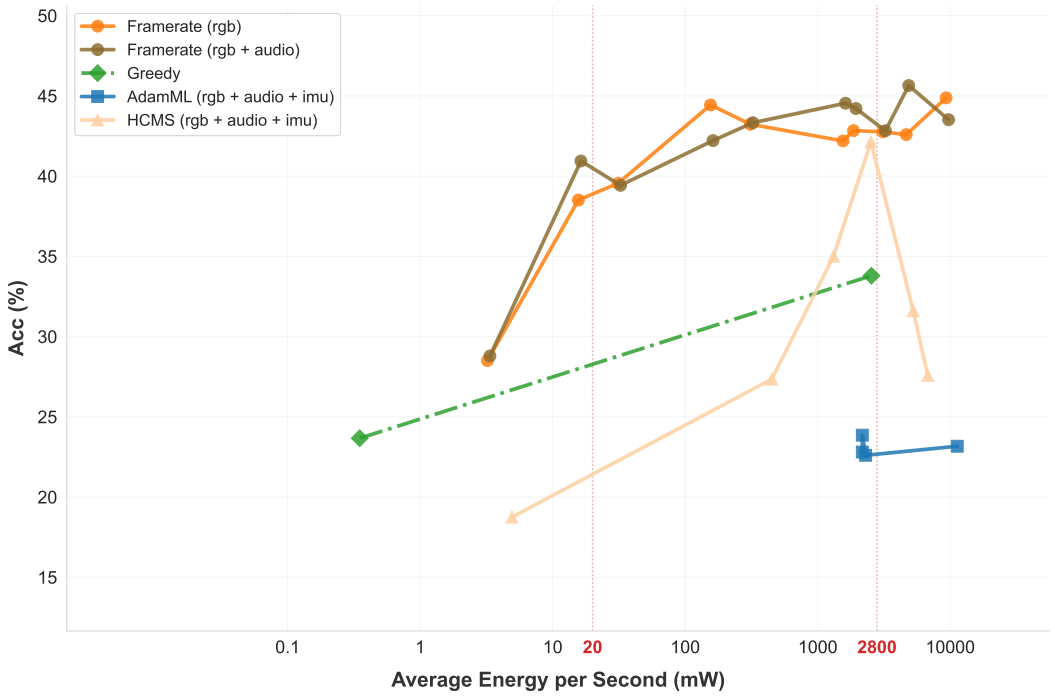


Figure 13: Energy-Accuracy trade-offs on Captain Cook 4D