Ego-Exo4D: Understanding Skilled Human Activity from First- and Third-Person Perspectives

Supplementary Material

Appendix

Table of Contents

Figure 8. The Project Aria device used for egocentric recordings.

7. Aria Glasses

For the Ego-Exo4D project, we chose to use Project Aria devices [38]. Project Aria is an egocentric recording device in glasses form-factor created by Meta. It is designed as a *research tool* for egocentric machine perception and contextualized AI research, and available to researchers across the world through projectaria.com.

7.A. Device and Sensors

The Project Aria device is built to emulate future AR- or smart-glasses catering to machine perception and egocentric AI rather than human consumption. It is designed to be wearable for long periods of time without obstructing or impeding the wearer, allowing for natural motion even when performing highly dynamic activities —such as playing soccer or dancing. It has a total weight of 75g (compared to over 150g for a single GoPro camera), and fits just like a pair of glasses.

Further, the device integrates a rich sensor suite that is tightly calibrated and time-synchronized, capturing a broad range of modalities. For Ego-Exo4D, *recording profile 15* is used, which uses the following sensor configuration:

- **One rolling-shutter RGB camera** recording at 30 fps and 1408×1408 resolution. It is fitted with an F-Theta fisheye lens that covers a field of view of $110°$.
- **Two global-shutter monochrome cameras** recording at 30 fps and 640×480 resolution. They provide peripheral vision, and are fitted with F-Theta fisheye lenses that cover a field of view of $150°$.
- **Two monochrome eye-tracking cameras** recording at 10 fps and 320×240 resolution.
- **An array of seven microphones** recording spatial audio around the wearer.
- **Two IMUs** (800 Hz and 1000 Hz respectively), **a barometer** (50 fps) and **a magnetometer** (10 fps).
- **GNSS and WiFi** scanning were disabled for Ego-Exo4D for privacy reasons.

All sensor streams come with metadata such as timestamps and per-frame exposure times. All data is made available in raw form as part of the Ego-Exo4D dataset. For convenience, we also include pre-computed slices of data that suit specific purposes, e.g., 2D gaze points, mp4s of each camera, and smaller .vrs files with a subset of sensor streams.

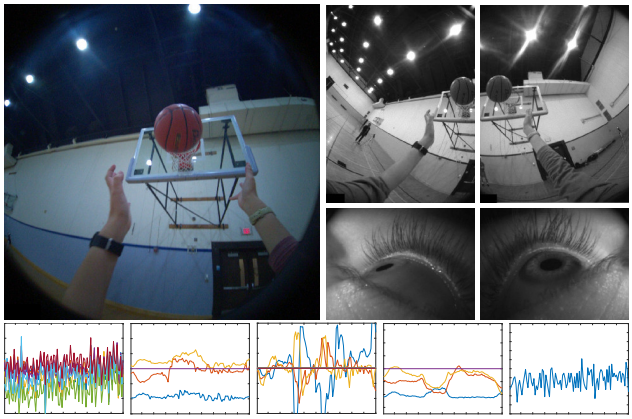


Figure 9. Sensor streams recorded by the Project Aria device. Top: RGB camera, left and right monochrome and eye cameras. Bottom: 10-second extracts from microphones, accelerometer, gyroscope, magnetometer and barometer respectively.

7.B. Machine Perception Services (MPS)

Project Aria's machine perception service (MPS) provides software building blocks that simplify leveraging the different modalities recorded. These functionalities are likely to be available as real-time, on-device capabilities in future AR- or smart-glasses. We use the following core functionalities currently offered by Project Aria, and include their raw output as part of the dataset. See [38] and the technical documentation for more details.

Calibration. All sensors are intrinsically and extrinsically calibrated. MPS also provides time-varying online-calibration that corrects for tiny deformations due to temperature changes or stress applied to the glasses frame.

Aria 6 DoF Localization. Every recording is localized precisely and robustly in a common, metric, gravity-aligned coordinate frame, using a state-of-the-art VIO and SLAM algorithm. This provides millimeter-accurate 6 DoF poses for every captured frame, as well as high-frequent (1 kHz) motion in-between camera frames.

Eye Gaze. The gaze direction of the user is estimated as a single outward-facing ray anchored in-between the wearer's eyes. We use an optional eye gaze calibration procedure, where the mobile companion app directs the wearer to gaze at a pattern on the phone screen while performing specific head movements. This information was then used to generate a more accurate eye gaze direction, personalized to the particular wearer.

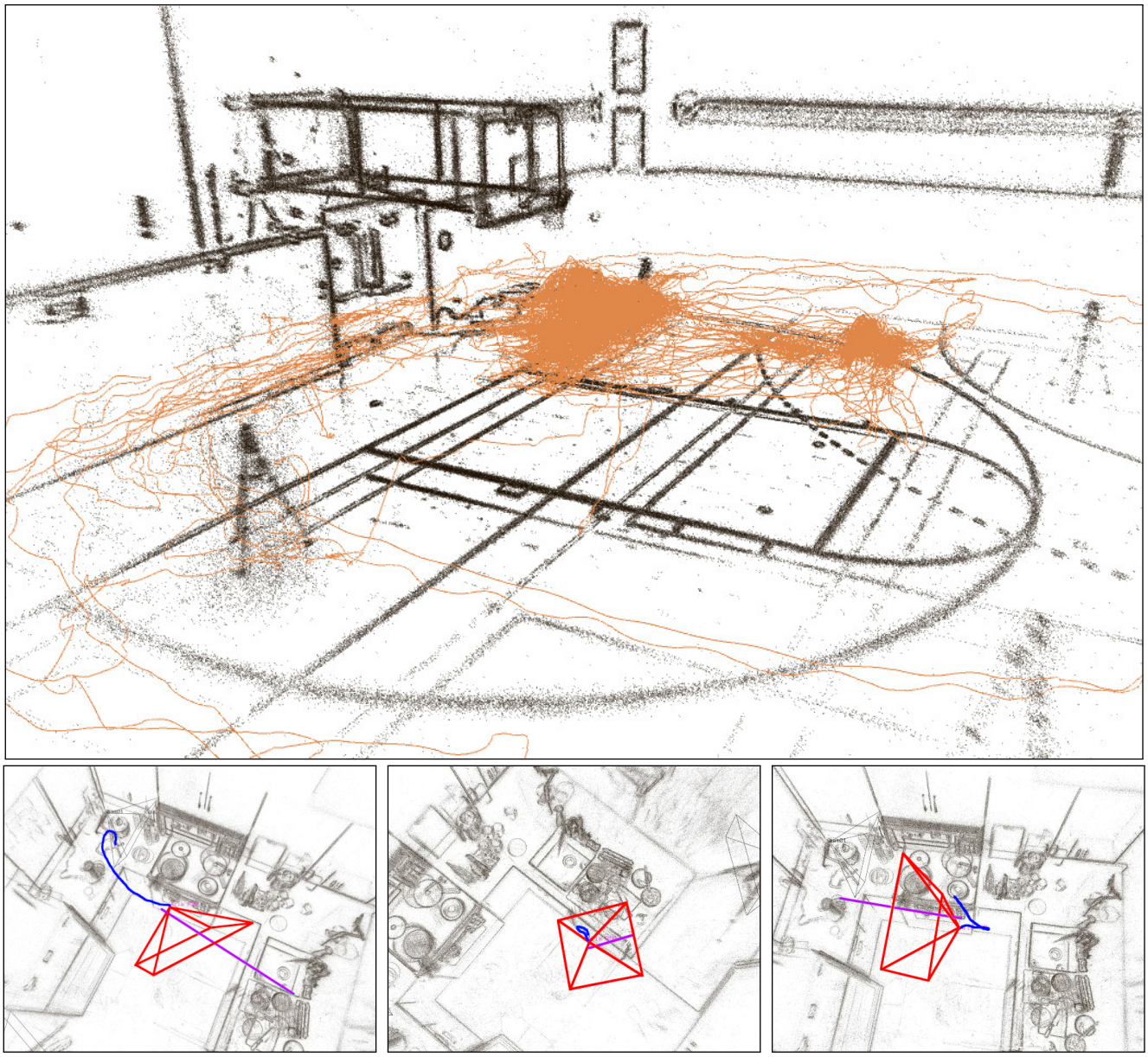


Figure 10. Aria MPS output for several recordings. Top: point cloud and estimated egocentric camera trajectory for a basketball session in Chapel Hill. This single continuous recording is 60 minutes long, has a total trajectory length of 2188 m, and contains 41 distinct takes. Bottom: three screenshots of a cooking recording, visualizing the current camera pose (red), eye gaze (purple), and last second of motion (blue).

Point Clouds. A 3D point cloud of static scene elements is triangulated from the moving Aria device, using photometric stereo over consecutive frames or left/right SLAM camera. The output contains both the 3D point clouds as well as the raw, causally computed, 2D observations of every point in the camera images.

GoPro 6 DoF Localization. For Ego-Exo4D, we added additional functionality on top of the existing Aria MPS functionality, specifically to localize the static GoPro cameras. To achieve this, we use the map built with Aria's SLAM cameras, and perform 6 DoF localization of GoPro frames on the map. To obtain the GoPro calibration, we manually calibrated one device in the lab to obtain default parameters, and then use the P4P [74] algorithm (with RANSAC to reject matching outliers) to estimate the 6 DoF pose, as well as re-estimate the focal length to compensate for possible calibration variation between devices.

7.C. Processing Summary

First, the MPS pipeline is invoked for *each full Aria recording*—these typically are about 20 minutes to 1 hour long and can include several takes, the hand-over in-between takes, as well as some other set-up steps. This is followed by localizing all GoPro videos of that scene as described above, and finally followed by time-synchronization across Aria and the GoPro cameras, as well as take-separation, as described below in Appendix 8.

There are total of 783 Aria recordings processed by MPS—containing the total 5,035 takes in the dataset. 95.9% of these recordings have successful Aria localization throughout the whole recording, with only 3.5% containing a partial tracking failure (leading to short gaps in the 6DoF trajectory). Three (0.6%) recordings failed completely. The most common failure reason is physical shock on the glasses, for example when the glasses are accidentally dropped on the ground or the table.

Furthermore we attempted to localize a total of 3,724 GoPro recordings, 91.4% of which are successfully localized. Similar to the Aria recordings, GoPro's are localized on a *recording* level rather than on a *take* level. This helps in particular with very short takes as are common during physical activities—as there otherwise would not be sufficient visual overlap across Aria and GoPro perspectives. The most dominant reason for GoPro localization failure occurs when the GoPro is pointed to an texture-less area (e.g. a white table) which lacks the necessary visual features to perform localization. As the GoPro's are static, this cannot be compensated for by device motion as is the case for the moving Aria device.

7.D. Tools and Ecosystem

Technical documentation and open-source tooling for Aria recordings and MPS output is available on Github[2] and the associated documentation page[3]. It includes both python and C++ tools to convert, load, and visualize data; as well as sample code for common machine perception and 3D computer vision tasks.

Contribution Statement

Jing Dong and Vijay Baiyya were responsible for obtaining camera poses, calibration, pointclouds and eye gaze using Aria MPS, created the 3D/4D visualizations for the paper and supplementary material, and acted as main contact points from the Aria team throughout the program; with Jing leading the algorithm development and verification, and Vijay leading the Aria MPS workflow and infrastructure development. Jakob Engel acted as technical and scientific advisor, and led the team that built the Aria Localization and Point Cloud algorithms. Kiran Somasundram helped design the capture setup and time-synchronization. Xiaqing Pan helped to align the Aria engineering team to support the EgoExo4D project. Mingfei Yan, Prince Gupta, and Sach Lakhavani acted as product managers of Aria and organizational leads for the successful use of Aria in the program. Kelly Forbes helped setting up agreements and working through the legal requirements of using Aria devices for recording the EgoExo4D dataset across the globe. Richard Newcombe initiated the Aria/Ego4D collaboration and acted as a scientific advisor throughout the program. Furthermore, we want to acknowledge the contribution of the entire Project Aria team as listed in [38], including Carl Ren and Sean Diener leading the Aria software and hardware engineering organization, and Renzo De Nardi as technical lead for the Aria device.

[2] https://github.com/facebookresearch/projectaria_tools

[3] https://facebookresearch.github.io/projectaria_tools/docs/intro

8. Camera Rig and Data Processing

The collection of ego-exo data at a global scale required us to develop a low-cost camera recording rig that was portable, auto-synchronized, and available internationally.

8.A. Hardware

Our unified camera rig is as follows: 1 Aria, 4 GoPros[4], 1 GoPro Remote, 4 Tripods, 4 SD Cards, 4 Tripod Mount Adapters, 4 Velcro'd Battery Packs, 4 USB-A to USB-C Cables, 1 Glasses Sports Strap, 1 Smartphone, 1 Laptop or Tablet for questionnaires. The total cost excluding the Aria/phone/laptop is under $3,000, with the majority of that going to the GoPros.

8.B. Time Sync

To sync cameras, we employ a pre-rendered sequence of QR Codes (*i.e.*, QR code video) that encode a wall-clock time. We show this QR code video using the smartphone at 29fps to all cameras in sequence and exploit the difference in frame rates to finely sync the cameras. In theory, the QR code decoded on a frame that captures a QR change is likely the one that was visible during that frame's center of exposure. With a single QR, the camera's center of exposure time could be anywhere within the 34.48ms that the QR is shown. However, with two consecutive frames with the same QRs, we can localize that time down to ± 0.574ms. The same approach yields ± 0.558ms for the 59fps GoPros given 3 consecutive frames (see Figure 11), providing sub-frame synchronization accuracy.

We manually verified that each GoPro camera was within 1 frame (+-16.66ms) of the Aria RGB camera by visually comparing them at single-frame moments (e.g., contact frames) using a synced video collage at the start and end of each capture. We checked points near the start and end of each capture under the logic that sync is a linear mapping and camera clock speed is mostly constant, so if the error is +- 1 frame at the start and +- 1 frame at the end, it will be +- 1 frame throughout.

An 'audio sync' fingerprint was played at the start and end of each capture to synchronize audio streams but has not been used.

Challenges and workarounds In practice, $\sim$70% of recorded captures yielded frame-accurate sync through our automated pipeline. Inaccurate sync causes included observed issues (*e.g.*, phone changing orientation mid-playback, video playback interruptions) and suspected ones (*e.g.*, videos not playing back at precisely 29fps, center exposure times not being evenly spaced). To recover these captures, we employed a manual sync procedure wherein people manually selected frame timestamps that should be aligned based on precisely time-localizable events, *e.g.*, a lighter first sparking, a soccer ball making contact with a cleat, or a hand beginning a fast slide down the neck of a guitar. This unblocked the remaining $\sim$30% of captures at the cost of less accurate sync.

[4]This represents the common core of the collection rig used in all capture settings. In certain captures, *additional* exo or ego GoPros are also used, as noted in Appendix 10.

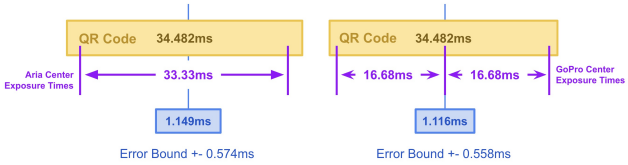


Figure 11. With a QR code timer playing back at exactly 29fps, cameras with evenly spaced center-exposures can be precisely time-localized to the QR timer with these multi-QR patterns.

Alternatives We explored and disqualified other sync options—notably using Timecode with TentacleSync or Ultrasync. Both of these solutions use LTC to encode a 1fps timestamp into the audio channel of a connected device. Using them with GoPros would cost us the stereo-audio modality, which we opted to keep to support audio-based research areas. We additionally lacked an ergonomic input solution for Aria to use while recording, so that mandated non-intrusive sync solutions.

8.C. Take Separation

To amortize the setup and tear down time required for each recording, we record multiple 'takes' (*i.e.*, one instance of a certain task) back-to-back and use a 'Take Separator' QR code (different from the time sync QR code video) that is identified in post-processing to auto-separate each take. This enables us to scale up recording—particularly for the physical scenarios where a single take can be less than a minute long. Data collectors track metadata for each take, identifying them by index and marking data such as participant ID (anonymous unique identifier), task (*e.g.*, making tea, making cucumber salad, performing CPR), and whether the take should be dropped (*i.e.*, if it is just setup time between activity enactments).

8.D. Recording Procedure

Our rig setup procedure entails setting up the stationary exo cameras in the recording environment and displaying QR codes to perform time sync and then take separations. Figure 12 overviews our recording procedure.

1. Position tripods, power on GoPros, and set camera angles to ensure maximum human coverage.
2. Begin Aria recording via smartphone.

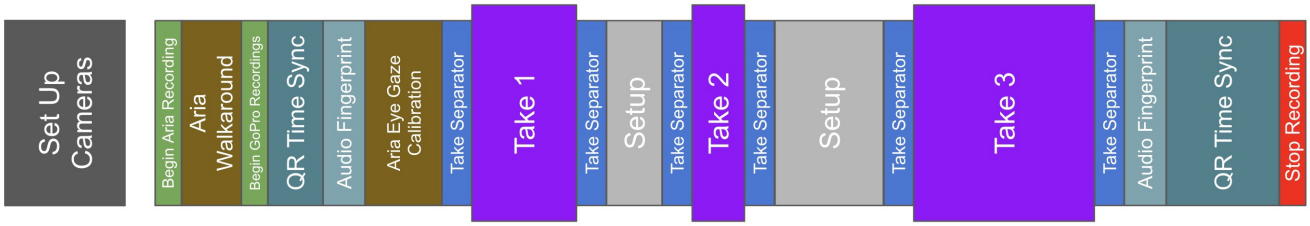


Figure 12. Overview of the recording procedure

3. Conduct a walk-around with the Aria glasses to build a basemap for 3D reconstruction and camera localization. Match the viewpoint of each GoPro camera by positioning the Aria directly in front of its lens.
4. Start QR Timesync Video off-screen. Show QR video to Aria RGB camera.
5. Use GoPro Remote to begin GoPro recording. Show QR video to each GoPro camera. Play Audio Sync fingerprint from the center of the space.
6. Pass Aria glasses to (new) participant. Perform Eye Gaze Calibration via the Aria app. Show 'Take Separator QR' to one GoPro and begin the take. Show 'Take Separator QR' to one GoPro after the take is complete and repeat this step for each participant/take. Do not repeat gaze calibration if the participant has not changed.
7. Play Audio fingerprint from the center of the space. Restart the QR Timesync Video off-screen. Show it to the Aria RGB camera, then each GoPro. Stop recording on all cameras.

8.E. Extensions

The core camera rig was extended to handle onsite requirements and regional challenges. The team at Universidad de los Andes introduced a top-down (ceiling mounted) GoPro for dance, which was adopted by the team at the University of Pennsylvania with an overhead pole mount. The teams at University of Pennsylvania, IIIT-Hyderabad, and Indiana University added an additional egocentric, head-mounted GoPro.

Contribution Statement

Rawal Khirodkar proposed the hardware and software specifications for the ego-exo camera rig and helped design the capture protocol. Sean Crane investigated various hardware setups leading to the final rig configuration and helped draft the capture guidelines including recommended gear. Devansh Kukreja developed the sync and take separation algorithms, experimented with different equipment options (e.g., camera, timecode boxes, mount options), and designed the interface to transfer data; he also managed the ingestion pipeline, the collaboration with Aria, the integration of their code for EgoExo, and usage of the .vrs files.

Procedural		**Physical**	
Cooking:	*Health*:	*Music*:	*Bouldering*:
- Omelette	- COVID test	- Violin	- V0 through V10
- Scrambled eggs	- Cardiopulmonary	- Piano	
- Tomato and egg	Resuscitation (CPR)	- Guitar	
- Sesame-ginger Asian salad			
- Greek salad	*Bike repair:*	*Basketball:*	*Soccer*:
- Dumplings	- Remove/install a wheel	- Mikan layup drill	- Freestyle dribbling
- Noodles	- Replace an inner tube	- Righthand reverse layup	- Freestyle juggling
- Pasta	- Clean and lubricate the chain	- Mid-range jump shot	- Penalty kicks
- Sushi roll	- Adjust rear		
- Samosa	derailleur (both limit		*Dance*:
- Coffee latte	screws & indexing)		- Easy choreography
- Chai tea			- Advanced choreography
- Milk			
- Cookies			
- Brownies			

Table 1. Specific activities collected for the three Procedural and five Physical domains

Dataset	**Year**	**Modalities**	**#Subj.**	**#Scenes**	**#Tasks**	**#Actions**	**#Masks**	**#BP**	**#HP**	**Nar.**	**EC**
Multimodal Egocentric Datasets											
EGTEA-Gaze [81]	2018	V,A,G	32	1	7	106	15k	-	-	✗	✗
MECCANO [126]	2021	V,D,G	20	2	1	61	-	-	-	✗	✗
EK100 [29]	2022	V,A	37	45	N/A	(97;300)∗	-	-	-	✓	✗
Ego4D [47]	2022	V,A,3D,S,G,I	931	74	N/A	110†	-	-	-	✓	✗
Multiview Datasets											
IXMAS [169]	2006	V	10	1	N/A	11	-	-	-	✗	✗
MEVA [26]	2021	V,T,GPS	100	28	N/A	37	-	-	-	✗	✗
Ego-Exo Datasets											
CMU-MMAC [77]	2009	V,A,M,I	43	1	5	-	-	-	-	✗	✗
Charades-Ego [144]	2018	1	71	N/A	N/A	157	-	-	-	✗	✗
H2O [76]	2021	V,D	4	3	N/A	36	-	-	0.5M	✗	✗
Assembly101 [109, 139]	2022	1	53	1	**101**	(24;90)*	-	-	0.2M	✗	✗
EgoExo4D	2024	V,A,I,G,3D,6D,B,Ma	**740**	**123**	43‡	**689**	**2.2M**	**9.6M**	**4.4M**	✓	✓

Table 2. Comparison between Ego-Exo4D and relevant datasets. Compared to existing datasets capturing both egocentric and exocentric views, Ego-Exo4D features more modalities, more subjects, and significantly larger scene diversity, as well as rich annotations including key-step segments, object masks, and three meticulously synchronized natural language descriptions paired with the videos (narrations, narrate-and-act, and expert commentary). To our knowledge, Ego-Exo4D also offers the largest available manual ground truth egocentric body pose annotations to date (in the above datasets or any others), and it has ∼14M total frames of 3D pose annotations and pseudo-annotations. *#Tasks* denotes the number of tasks that subjects were asked to execute in each dataset, *Subj.* denotes recorded subjects, *#BP* refers to number of 3D body poses, *#HP* refers to number of 3D hand poses, *Nar.* denotes narrations, and *EC* refers to expert commentary annotations. Modality abbreviations: **V**ideo, **A**udio, **D**epth, **G**aze, **S**tereo, **IMU**, **3D** Environments, **T**hermal IR, **GPS**, **M**otion Capture, **6D**OF, **B**arometer, **Ma**gnetometer. * denotes action taxonomies defined in terms of verbs and nouns, statistics reported as (number of verbs; number of nouns). † The number has been taken from the Moment Query benchmark. ‡ Number of tasks for Ego-Exo4D includes 21 procedural activities and 22 physical activities (listed in Table 1).

9. Scenarios

Ego-Exo4D focuses on eight domains of human skill. See Table 1. These were selected to capture diversity, while also scoping our effort sufficiently to build a density of data within a core set of skills. Our scenarios were further targeted after a review of existing activity datasets, consideration of priority elements for transfer learning, and the identi-

fication of entry points to capture real-world scenarios available uniquely to our partners. To support data collection, we narrowed our scope to a small subset of tasks within each domain.

As discussed in the paper, we divide these into the categories of *procedural activities*—requiring a sequence of steps to achieve a goal state—and *physical activities*—requiring some degree of refined physical control to perform appropriately, but not necessarily following a step-by-step order. Note that in general an activity can be both procedural and physical; they are not mutually exclusive properties. Each consortium member selected multiple domains of focus from the eight targeted domains (cooking, health, etc.) based on their own preferences and local opportunities to capture data of these scenes at scale.

In working groups, our consortium developed data collection guidelines for each of our eight domains. These guidelines described the recommended camera positioning, instructions for participating camera wearers, along with important context-specific considerations. For example, given privacy concerns, our health guidelines required data collection participants to discard COVID tests before results were visible. These guidelines provided general parameters from which to collect data; however, they were not rigid steps.

To support diversity and implementation at a global scale, we allowed for site-specific adjustments to be made. For this reason, our dataset contains important nuances that should be noted, including, for example, differing standards for the implementation of CPR, cultural differences in the ingredients used for different targeted dishes, and location-specific bouldering routes.

Additionally, while data collection partners were encouraged to select scenarios of their preference, we did set broad targets for each site to collect at least three domains among the eight options. Since every domain is covered by more than one partner, the dataset exhibits visual variety from the different physical locations. Even within the captures done by a single partner, there are often multiple different sites used for filming (e.g., a couple different bike shops in the same city).

These domain selections were additional to our one cross-cutting scenario of cooking, which was collected by all partners. Cooking was a priority domain because it resonates around the world as a human need and interest. We are pleased to report that the cooking scenario of Ego-Exo4D contains more than 650 takes of cooking performed by more than 170 chefs in 60 different environments around the world, forming nearly 100 hours of ego video alone.

Figure 13 and 14 shows example frames illustrating the variety of the sites and tasks.

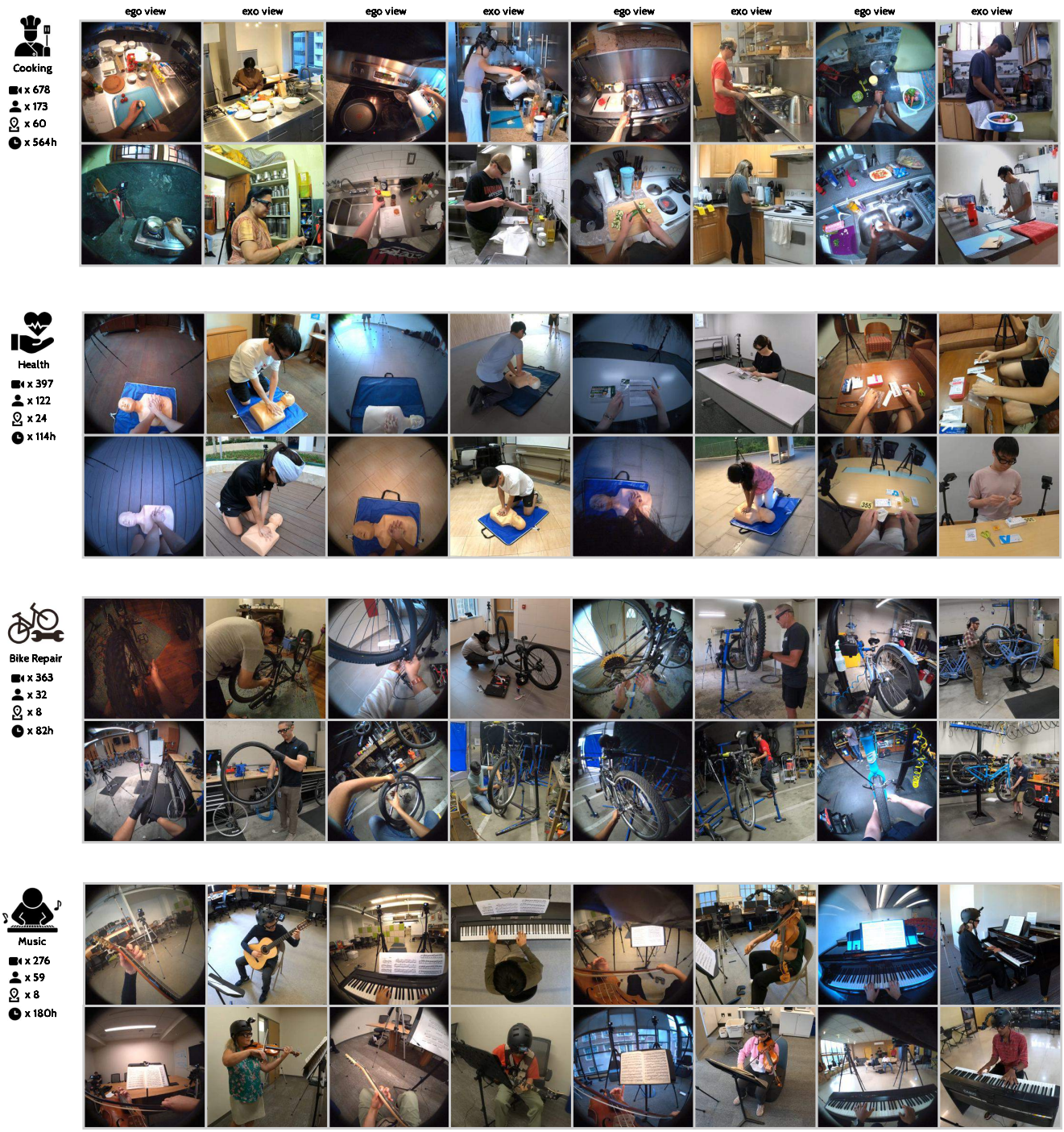


Figure 13. Ego-Exo4D captures skilled activity from 8 domains, in a wide variety of 123 scenes in 13 different cities in Japan, Colombia, Canada, India, Singapore, and seven US states. Every odd column shows an ego view, and the adjacent even column shows one of its paired exo views.

Figure 14. Ego-Exo4D captures skilled activity from 8 domains, in a wide variety of 123 scenes in 13 different cities in Japan, Colombia, Canada, India, Singapore, and seven US states. Every odd column shows an ego view, and the adjacent even column shows one of its paired exo views.

Contribution Statement **Bike Repair** Yale Song led the scenario development and collection guidelines for Bike Repair. Jim Rehg also contributed to the working group.

Culinary Manolis Savva led development of the scenario and collection guidelines. Other contributors in the working group: Andrew Westbury, Kumar Ashutosh, Deepti Ghadiyaram, Gene Byrne, Kristen Grauman, Santhosh Kumar Ramakrishnan, Shun Iwase, Yan Xu

Health Mike Zheng Shou was the lead author of the health scenarios guidelines.

Bouldering and Dance Pablo Arbeláez and Maria Escobar were the lead authors of the Bouldering and Dance guidelines.

Music Jianbo Shi and the team from the University of Pennsylvania were the lead authors of the Music guidelines.

Basketball Gedas Bertasius was the lead author of the Basketball guidelines.

Soccer Rawal Khirodkar was the lead author of the Soccer guidelines, camera placement, size of the capture area, and drills that were captured.

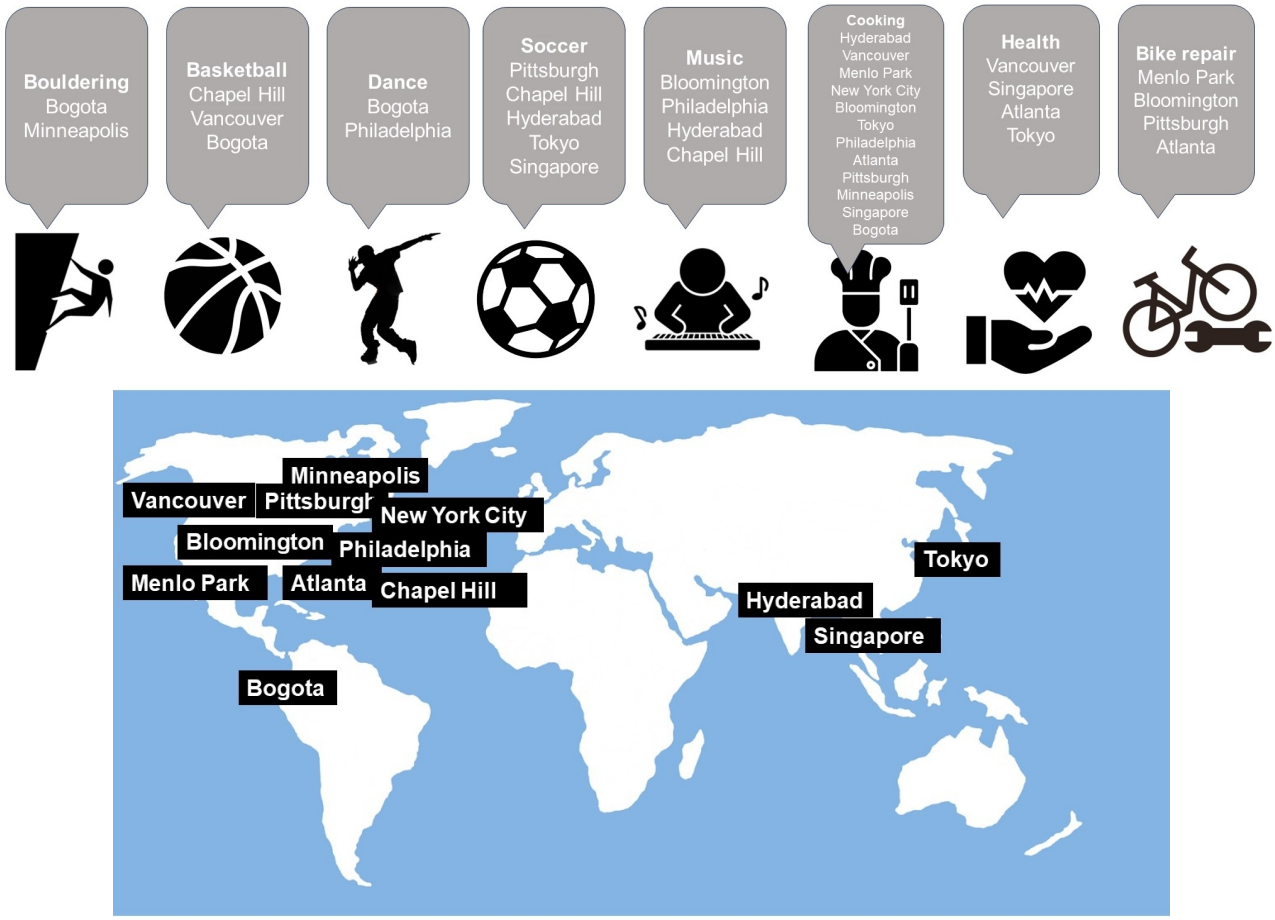


Figure 15. Geographic coverage of Ego-Exo4D and breakdown of which scenarios are captured in which cities. Note that even within a given city, there may be multiple sites (e.g., multiple bike repair shops or kitchens in the same city).

10. Data Collection

Twelve research labs came together for nearly two years to create Ego-Exo4D. Importantly, our collection across the sites was a coordinated effort, with common guidelines, scenarios, and camera rigs. In this way, the dataset is cohesive at the same time it is diverse. In this section we describe the data collection details that are specific to each partner site, e.g., how they recruited participants, which of the 8 scenarios they captured, or any modalities they added on top of the common rig.

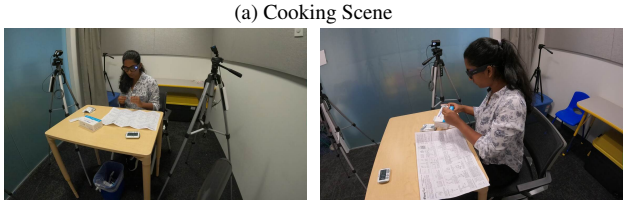
Figure 15 shows the breakdown of which scenarios were captured by each partner institution as well as a map highlighting the locations of the 12 labs involved in data collection. Note that an additional four institutions not shown on the map are part of the consortium (e.g., contributing to benchmarks) but did not collect data. They are UT Austin (USA), KAUST (Saudi Arabia), University of Catania (Italy), and University of Bristol (UK).

10.A. Carnegie Mellon University

Carnegie Mellon University focused on three skill-based activity scenarios: (1) soccer, (2) bike-repairs, (3) cooking. The exocentric cameras for our collections, four in total, were arranged approximately in a square configuration at a consistent height to capture the full range of the activity. Notably, for the soccer activities, an additional exocentric viewpoint was positioned inside the goal post to offer a more comprehensive perspective on the participants.

Soccer In the soccer scenario, we collaborated with professional players from the Pittsburgh Riverhounds team, representing the experts, and students from Carnegie Mellon University (CMU) as the beginners. We captured the soccer scenario across 4 different locations. The drills featured a variety of movements such as dribbling, goal kicks, and juggling, with each participant performing for a minimum of 3 minutes. This scenario resulted in roughly 4 hours of egocentric footage and 18 hours from exocentric perspectives, encompassing 32 participants in total.

(a) Cooking Scene



(b) COVID Test Scene

(c) Bike Repair Scene

Figure 16. Views from two different cameras for each scenario collected in Atlanta, GA, USA.

Bike repair In the bike-repair segment, our experts were seasoned mechanics with over a decade of experience from Allegheny county. To ensure authenticity, we visited each mechanic in their respective shops to allow usage of their own tools and setup. Four tasks were captured for each bicycle, and we ensured bicycle diversity by selecting different sizes, shapes, colors, and makes. The tasks include tire removal, tube change/ inflation, tire reassembly, and clean/lube chain. This yielded 3 hours of egocentric recordings and 12 hours of exocentric footage, encompassing 22 different bicycles.

Cooking For the cooking section, we documented a professional chef in his traditional kitchen environment. Our dish of choice was scrambled eggs, and to inject variety, the chef prepared it using different techniques. This segment summed up to an hour of egocentric recordings and 4 hours from exocentric viewpoints.

All recordings were conducted in Pittsburgh, PA, USA, strictly adhering to CMU's Institutional Review Board (IRB) guidelines. Every participant was briefed about the recording process, and prior to their involvement, a signed consent form was obtained.

10.B. FAIR, Meta

We collected 119 total takes of skills demonstrations in New York and three different locations in California. We focused on cooking and bike repair, taking advantage of the skilled workforce of chefs and bike technicians that serve our campuses and employees in these locations. We used the unified camera rig of 1 Aria and 4 GoPros without any additional sensors.

Bike Repair Our skilled mechanics performed four different bike repairs for a total of 102 takes. We focused specifically on wheel repairs (removing and installing the wheel & flat repairs). While we strive for diversity in terms of the model of bikes, a majority of those in the dataset are drawn from standard fleet bike models, which contain identical parts and components. The location featured in the dataset is a well-equipped, industrial scale bike shop.

Cooking Our chefs recorded five different recipes as part of 17 unique takes, including salads, egg dishes, and Asian Garlic Noodles. Locations featured in the dataset are three different professional kitchens used to prepare and serve hundreds of employees each day.

Internal documentation and processes ensured all participants provided informed consent to appear in the dataset and participation was strictly voluntary.

In total, we were able to mobilize five chefs and four bike mechanics. Due to workplace considerations, we did not collect demographic or age information about our chef and mechanic partners. Participating chefs and bike technicians are highly skilled with all research subjects reporting that they do the activity shown in the dataset daily or weekly. Similarly, eight research subjects have more than 10 years professional experience.

10.C. Georgia Tech

Collection at Georgia Tech focused on the Health, Cooking, and Bike Repair scenarios. Across these 3 scenarios, 279 takes were captured with 34 unique participants. For all scenarios, the unified camera rig was positioned such that 2 exocentric cameras would ensure capture of the participant's hands, and the other 2 exocentric cameras would capture the participant's full body and the full environment.

Participants were recruited from different sources including flyers, campus organizations, email lists, and word of mouth. Five of these participants completed data collections for 2 scenarios (4 participated in Health and Cooking, and 1 participated in Cooking and Bike Repair). Potential participants were provided with the study description and consent form prior to scheduling a recording session. At the beginning of each session, study personnel walked through the consent form with the participant, and answered any questions. The participant then reviewed and signed the consent form to confirm participation in the study.

The recording environment differed by scenario and included participants' homes, campus meeting rooms, and an on-campus bike shop. Fig 16 shows a sample environment and camera setup for each of the Health, Cooking, and Bike Repair scenarios. Further details of the data collection specific to each scenario is provided below.

Health Participants for the Health Scenario took COVID rapid test kits while seated at a table. Recordings were captured in 2 different on-campus meeting rooms. Participants were recruited through campus email lists and flyers in local coffee shops. Each recording session lasted approximately 40-60 minutes and consisted of a participant completing 5-7 test kits, using 2-4 different types of test kits. 7 different types of COVID test kits were used across the full collection. In total, 96 takes were recorded from 16 unique participants.

Cooking Participants for the Cooking Scenario prepared dishes from three recipes: Asian Salad, Tomato & Eggs, and Garlic Noodles in their home kitchens. Participants were recruited via mailing lists of local apartment complexes, contacting participants from prior research studies, and word of mouth. Each recording session lasted 2-3 hours, capturing 3-6 takes of a recipe being cooked from start to finish. Participants cooked 2-3 of the recipes during their session, depending on dietary restrictions and preferences. Participants were provided with ingredients and a paper copy of the recipe, and used equipment from their own kitchen to prepare the food. In total, 71 takes from 15 unique participants were captured. The takes were about evenly distributed among the three recipes. Recordings were completed in 10 unique kitchen environments.

Bike Repair Participants for the Bike Repair Scenario performed repairs including taking off a wheel, putting on a wheel, replacing a tube, and cleaning a dirty chain. We recruited skilled participants from a campus bike repair organization. There were 8 unique participants, who each completed 1-3 recording sessions. Each session lasted 40-60 minutes and captured 5-7 takes of individual bike repairs. In total, 112 takes were recorded. showing the distribution across repair tasks. One session of 6 takes was recorded in a participant's home, while the rest were recorded in the campus organization's bike shop space, which is shown in Fig 16c. Due to the organization's access to a large quantity of used bicycles, there is large diversity in the make and model of bicycles across takes.

The study protocol was reviewed and approved by our Institutional Review Board (IRB).

10.D. IIIT-Hyderabad

In Hyderabad, we contributed to three scenarios - (a) cooking, (b) soccer, and (c) music. We formulated a data collection strategy tailored to the specific scenario, as outlined below.

Our primary objective was to comprehensively capture body and hand movements, along with their interactions with objects, during the execution of the activities. In general, we adhered to the standard camera setup instructions.

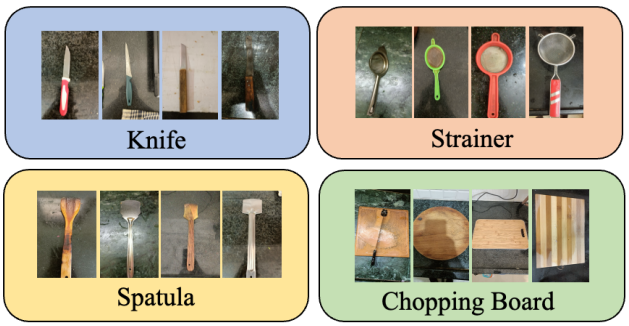


Figure 17. In Hyderabad, India, cooking was captured in different kitchens with socio-economic diversity. We observe that the same kitchen tools appeared in different shapes.

Nonetheless, we incorporated an extra exo-camera for capturing soccer activities in order to enhance the overall coverage of the event. Additionally, for music activities, we introduced a head-mounted Go-Pro camera.

This decision stemmed from the observation that expert musicians frequently do not directly look toward their instruments while playing. Consequently, the head-mounted camera guarantees continuous visibility of both the hands and the musical instruments, providing an ego-view perspective.

The collection in India was done during the peak summer, and this led us to a challenging situation where the cameras frequently shut down due to overheating. To address this, we mostly avoided capturing multiple takes in one capture and placed the cameras into an ice chest box in between the captures to cool them down.

Cooking For cooking, we reached out to people located in Hyderabad with varying socio-economic backgrounds and explained the data collection plan, and goals. We also requested them to engage their family members as well as friends in this data capturing process. Finally, we recorded the videos with the 41 informed participants capturing in a diverse set of 19 kitchens, geographically well-apart in and around Hyderabad, resulting in a rich dictionary of kitchen utensils (see Figure 17), narrations in four different languages. Additionally, we made an effort to ensure a balanced representation of genders in our overall data collection process.

Soccer For soccer, we reached out to three different soccer training schools in Hyderabad with the overall recording plan and process. They helped us in recruiting local soccer teams who play professional tournaments and practice almost everyday. We also recruited few players from our university soccer teams. In total, we recorded 49 participants, 'performing dribbling, juggling, and penalty-kick activities.

Music For the music scenario, we contacted one music school and recruited 4 musicians from them having at least 3 years of experience of playing either the piano, guitar,

or both instruments. To add diversity, the musicians were asked to play western as well as Indian pieces.

Our collection protocol was reviewed and approved by our university's Institutional Review Board (IRB). The primary conditions set forth by the IRB encompass the following aspects: (a) participants with 18+ age are deemed suitable for inclusion in the project, (b) participants have provided explicit consent for their facial and vocal presence to be featured in the released videos, (c) participants have willingly agreed to take part without receiving any immediate financial incentives from the videos, and (d) participants have the autonomy to engage in the activities in an environment of their choice. The participants were given the detailed descriptions of the project beforehand and requested to sign the consent form. Each participant received compensation as part of the process.

We selected participants from a wide range of age groups, spanning from 18 to 61 years old, to introduce an additional layer of diversity. Moreover, the participants were from diverse professional backgrounds (*e.g.*, coach, software engineer, data annotators, project managers etc.). Before sharing, we carefully examined each video to ensure there was no sensitive content.

10.E. Indiana University

We focused on cooking, bicycle maintenance, and music scenarios. All activities were collected using the unified camera rig, including additional sensors in specific scenarios. For cooking, the 4 GoPros were placed 90 degrees apart from each other, with 2 placed close to the participants to capture hands and objects and 2 placed further to capture the overall scene. In music, 4 GoPros were placed in front of the player, approximately 45 degrees apart from each other. In addition, we attached an additional GoPro HERO10 camera to the participant's head (using a helmet), tilted down roughly 80 degrees to capture hand movements. In bike repair environments, the 4 GoPros were placed 90 degrees apart from each other, of which 1 GoPro was placed close to the bike, 1 GoPro was placed close to the workbench and tools, and 2 GoPros were placed further away from the participants to capture the overall scene.

Cooking For cooking, we had a total of 18 participants collect 72 takes and 20.5 hours of video. For 15 of the participants, we used a commercial test kitchen at our university. We purchased all of the ingredients and kitchen equipment ahead of time and had them ready when each participant arrived. We asked them to make four dishes (chai tea, sesame-ginger salad, tomato and eggs, and noodles) and provided printed recipes for these dishes. The remaining three participants chose to record in home kitchens, and the four dishes they made varied based on their preferences (one participant made omelet, cucumber salad, noodles, and chai tea, another made scrambled eggs, sesame-ginger salad, sushi rolls, and brownies, and the third made scrambled eggs, cucumber salad, noodles, and milk tea). Due to concerns about food safety, we discarded (composted) the cooked dishes instead of allowing the participants to eat them.

Music For music, we had a total of 17 participants collect 60 takes and 6.5 hours of video. Participants were recruited based on their self-assessed proficiency in one of three instruments: piano, violin, or guitar. We recorded in 4 different locations including two studios, an office, and an auditorium that had a piano. Participants were instructed to play scales and arpeggios (2 mins), sheet music provided by us (3 mins), freeplaying (10 mins), and then recall and talk about any mistakes that were made during the playing and what could be improved (2 mins).

Bike repair For bike repair, a total of 13 participants recorded 108 takes and about 8 hours of video. We initially planned to hire professional bike technicians, but it was very difficult to recruit them in our relatively small city. Instead, we recruited more generally, looking for participants with (self-assessed) proficiency to do four basic bike maintenance tasks: removing a wheel, changing an inner tube, reinstalling a wheel, and cleaning and lubricating the chain. Most of the takes were recorded in a small house that is used for storage by our university's landscaping staff, and provided a realistic garage-like environment. We provided participants with a bike rack and supplies including bike tubes, pumps, tools, chain cleaner and lubricant, and gloves. To achieve diversity in different bikes and bike types, we asked participants to bring their own bike when possible, and we also provided 4 bikes (one of which belonged to one of the authors and the other three which we bought at a salvage shop). Most participants performed takes on about 3 bikes. One participant chose to record in an apartment, and one recorded in a hallway in a university building instead of the garage due to scheduling conflicts.

Our protocol was reviewed and approved by our university's Institutional Review Board. For each potential participant in each scenario, we first scheduled an online introduction meeting to tell them about the study and answer their questions and concerns. If they were interested, we agreed on the activity they would perform and when and where to meet for recording. We also sent them the informed consent form to give them sufficient time to review. On the recording day, we first asked them to sign the consent form, and then started recording their activities. All activities were recorded in an enclosed space to make sure that no one else accidentally entered the field of view of the cameras. We also ensured that the space did not have privacy-sensitive content, and we instructed participants not to use their phones or other devices that might show private content.

Within a few days, we securely sent the videos to the

participant so that they could review the video and ensure that they were comfortable sharing it with others. They also completed a brief online demographic study, and then were sent an incentive payment in the form of an electronic Amazon.com gift card. We made clear to participants that if they were not comfortable sharing their video, we would destroy it and they would still receive their incentive payment, although none of the participants chose this option. We gave the participants US$20 in gift cards for each hour of their time spent recording (with a minimum of $20, and partial hours rounded up to the nearest $5). We gave an additional $20 gift card to reimburse travel costs for those who came to our facilities to record (e.g. in our kitchen, bike repair shop, or on-campus studio or auditorium). For cooking and bike repair, we gave an additional $20 gift card to participants who provided their own ingredients or bike maintenance supplies, to defray these expenses.

We recruited participants in the Bloomington, Indiana, USA area through online email advertisement, word of mouth, physical flyers, and posting on social media. We recruited participants who were 18 years of age or older, had self-assessed expertise in the activities as described above and could perform the tasks without wearing prescription glasses (which could interfere with the Aria's gaze tracking).

10.F. National University Singapore

In Singapore we focus on the following scenarios: soccer, health-related activities including COVID-19 ART testing and Cardiopulmonary Resuscitation (CPR), and cooking. In total, our collected data encompasses around 26 hours of egocentric videos and 117 hours of exocentric videos. These videos spread across 327 takes. In general, we adhered to the standard camera placement guidelines; however, for each scenario, we fine-tuned the position of the exo cameras based on practical considerations. For instance, in a small kitchen for cooking, we positioned the camera on the table to broaden its field of view.

Soccer For soccer, we conduct recordings at a university sports field. Our participants were primarily sourced through referrals provided by skilled participants recruited through online calls for participants. Additionally, during outdoor recording sessions, we occasionally invited surrounding bystanders to participate.

Health For health activities, we recorded in vacant classrooms, meeting rooms, and outdoor fields. CPR sessions are captured either in a yoga classroom or in a quiet, empty outdoor field. For recruitment, we circulated online calls for participants and then, for skilled activities like CPR, we collaborated with experts to organize training courses. Participants would participate in these courses and were trained to be proficient and then conducted recording afterwards.

Cooking As for cooking, which requires a kitchen, we used the kitchen in our lab mates' apartments and arrange other participants to go there.

Our data capture has been approved by our university's Institutional Review Board (IRB). The main requirements include that participants: (1) agreed to take part in the study, (2) agreed to donate their speech, image, video, IMU, and 3D scan data for the purposes of this research, (3) agreed that their face, tattoos, and voice may appear in the data, (4) have the right to withdraw their recorded data at any time.

In Singapore, high temperatures often pose the challenge of camera overheating, particularly for GoPro cameras, which can lead to protective shutdowns and interrupt data collection. To mitigate this, we place small ice cubes wrapped in wet wipes on the GoPro cameras to help cool it down during recording. Furthermore, we attempted to schedule our participants' recordings in the evening or during an overcast day.

Our data pool comprises contributions from about 93 meticulously selected participants, ensuring a proficient completion of the recordings. Particularly in soccer, most participants have extensive experience and were members of their school or college soccer teams.

10.G. Simon Fraser University

We captured three types of scenarios in a variety of environments: cooking, basketball, and COVID-19 testing. In total, 88 participants carried out activities in the three scenarios we collected in a total of 61 data capture sessions, resulting in 519 activity takes.

We used the unified camera rig and followed the general collection guidelines with a number of small adjustments to facilitate scenario-specific capture. In kitchen and health scenarios where the participant interacts with small objects in tabletop height settings, the placement of exocentric cameras was optimized in a "two near, two far" setup to provide for visibility of the small objects and hands while also capturing the overall human pose during the activity.

Cooking The cooking scenario was captured in a decentralized fashion by going to the participants' own residences and asking them to cook in their kitchen. This allowed for diversity in the environment as well as in the participant during data capture. Our data capture sessions resulted in 112 cooking takes.

Basketball Collection for the basketball scenario was done in a "round robin" fashion to reduce player-to-player overhead. We targeted a spectrum of experience levels, for example going from university basketball team players who compete at the national level to more amateur basketball players who only have played basketball occasionally. We collected 355 takes of basketball activities.

Health Following the standard data collection guidelines for health activities, we gathered 52 takes of health activities.

We followed the institutional research board (IRB) process at our institution to acquire approval for the participant recruitment strategy, study setup, and participant consent acquisition forms. All participants consented to their data being collected and distributed for research purposes. Participants have the right to request that their data be withheld from inclusion in the dataset.

We recruited participants by word of mouth, reaching out to specific clubs and groups for some of the activities, and more generally through advertisement using university-affiliated communication channels.

10.H. Universidad de los Andes

We collected around 40 hours of video spanning four distinct scenarios that encompassed three physical activities (basketball, bouldering, and dancing) and one procedural activity (cooking). Figure 18 shows examples of the diverse scenarios that we collected. In total, we collected 2062 takes across all the activities. We used the unified camera rig with additional activity-specific sensors as described below.

Bouldering We partnered with a local climbing gym, which serves as a teaching and competition center in Colombia. We used the gym as the recording location and recruited participants who practice or teach bouldering there. Our focus was to recruit participants with four different levels of expertise: beginner, intermediate, advanced, and professional climbers. We hired expert route setters to design 33 climbing routes. These routes varied from beginner (V1) to expert level (V7). For data collection, each participant attempted to complete seven routes, having 3 minutes to make as many attempts as possible for each route. The routes were selected considering the expertise level of each participant. We located four exo cameras to capture each take; two horizontal cameras were facing the climbing wall, and the other two vertical cameras were on each side of the wall. Thus, the four cameras captured a complete view of the climbing wall and the participant's movements at every moment. We gathered 1251 takes for the bouldering scenario from 40 participants. We ensured ethnic, age, and expert-level diversity across the takes.

Dancing We collaborated with a salsa dance academy to use as a recording location and to help with participant recruitment. We recruited students from three expertise levels: beginners, intermediate, and advanced. According to the expertise level, each dancer performed different choreographies. Beginners recorded a single choreography, while intermediate and advanced participants recorded an additional one according to their expertise. Each attempt lasted one minute, and the dancer performed from six to ten attempts. The choreographies were designed by professional dancers who teach at the academy. We used five exo cameras: four forming a square, defining the dancing area, and the fifth camera placed on the ceiling. Given the salsa dance's velocity and the movements' complexity, this fifth exo camera gave a crucial point of view for further analysis. We gathered 600 takes from 40 participants across the three expertise levels.

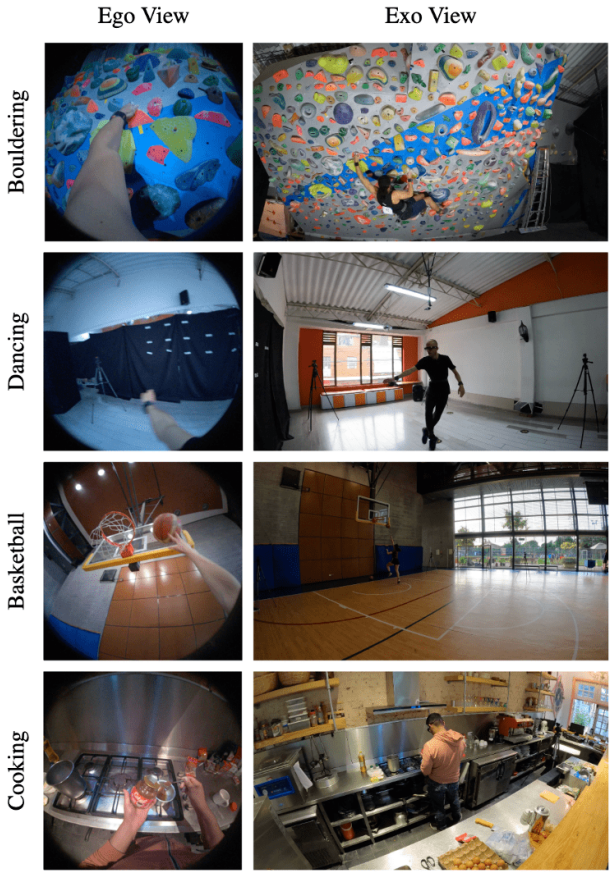


Figure 18. Egocentric and one exocentric view for each of the recorded scenarios in Bogota, Colombia.

Basketball We collected data from the professional women's team and students from a basketball class at our University. Each participant performed six to ten attempts for each basketball exercise. We collected all captures at the basketball court at our University's Sports Center. For this setup, we used four exo cameras around the basketball ring, ensuring a complete view of each exercise. For this scenario, we collected 167 takes from 38 participants.

Cooking We rented a professional kitchen equipped with all the necessary utensils to perform the captures. We focused on collecting data from two types of recipes: a dish with egg and a drink. Each participant could choose be-

tween cooking an omelet, scrambled eggs, tomato and eggs, and coffee latte or tea for the drink. Each participant was free to choose how to complete each recipe. Thus, our takes show diverse ways to prepare each recipe. For this setup, we placed four exo cameras around the kitchen, all facing the user, to capture the whole kitchen without losing any detail of the person making the recipe. We placed two cameras on a counter facing the kitchen and the other two on each side of the kitchen. We collected 44 takes for the cooking scenario from 20 participants.

The Institutional Review Board (IRB) of our university reviewed and approved our study protocol. All participants signed a consent form before participating in the study.

We partnered with professional training centers for physical activities that helped us recruit volunteers with different expertise levels. These volunteers were previously familiarized with the activities and the environment where the captures occurred. In addition, we recruited family members, friends, and acquaintances of students and faculty members of our research group for cooking.

10.I. University of Minnesota

Collection at the University of Minnesota, Twin Cities focused on two main scenarios: Bouldering and Cooking. A total of 249 takes with 53 unique participants were collected. We collected all data using the unified camera rig with no additions.

Bouldering The bouldering activity was collected at a local bouldering gym, focusing on a wall with 14 different routes ranging in difficulty from beginner to expert. We collected 210 takes from 42 unique participants. Participants were asked to climb four to five routes of their choice, with the ability to take breaks within or between takes. Expert climbers who felt comfortable with the routes were able to narrate their approach and climb in real time. As participants were able to choose routes freely, our five exo-cameras were set up to accommodate the entire wall.

Cooking Cooking activity was collected on-site at each individual's home kitchen. Five exo cameras were set up in each kitchen to maximize coverage of both the participant and the environment. We captured 9 unique kitchen environments with 14 unique participants whose skill levels ranged from cooking novice to commercial chef. Participants focused on three recipes each (scrambled eggs, Greek salad, and pasta noodles from scratch), which were performed back-to-back on the day of recording.

Our data collection protocol was reviewed and approved by the Institutional Review Board at our university. At every take, the study personnel provides a guidance to a participant through the consent form prior to participation, ensuring the participant understands the purpose of the study and all risks involved, with each participant receiving payment proportional to their contribution.

Participants were recruited via word of mouth, campus organizations, and digital flyers which were distributed via local social media (Facebook) communities.

10.J. University of North Carolina

Throughout our data collection at UNC, we focused on three skill-based activity scenarios: (1) basketball, (2) soccer, and (3) music drills. We used three unique environments (i.e., a basketball gym, a soccer field, and a music studio) to capture the data for each scenario. All recordings took place on the UNC campus. UNC's Institutional Review Board (IRB) reviewed and approved our study protocol. All participants signed a consent form before participating in the study.

To recruit participants, we used an online research study database, where participants from the local area could sign up to perform our study. We recruited participants willing to perform skill-based activities such as basketball, soccer, or music drills regardless of their skill level. Additionally, to recruit a more skilled group of participants, we contacted expert musicians from UNC's School of Music and athletes from UNC's basketball and soccer teams.

In total, we collected approximately 19 hours of egocentric and 76 hours of exocentric video data spanning approximately 548 takes of activity demonstrations from 56 participants (41 male, 15 female). Among the 56 participants, 44 were aged 18-25, 10 aged 25-50, and 2 aged 50-75. Furthermore, 26 participants had more than 10 years of experience in the scenario they chose to perform (e.g., basketball, soccer, music), 13 participants had 1-10 years of experience, and 17 had less than 1 year of experience. We used standard camera placement guidelines and the same recording devices described above.

Basketball All participants performed three basketball drills: Mikan Layup, Reverse Layup, and Mid-range Jumpshooting, for 388 takes. We recruited 11 expert players from the university team with 10+ years of experience. To improve the participant skill diversity in the dataset, we also recruited novice players with less than 1 year of playing experience. The location of data collection was a university basketball gym.

Music For the music scenario, we asked all 9 participants to play 5 minutes of scales and arpeggios and 10 minutes of free play for 27 takes. All of our participants were recruited from the university music club and considered themselves as experts at playing their respective instruments. The instruments featured in our collected dataset were piano, trombone, trumpet, and saxophone. The Ego-Exo4D music guidelines called for just piano, violin, and guitar, but we found it necessary to expand this list in order to gather data for this domain. All data was recorded in a

university music room.

Soccer For soccer, we focused on three drills: dribbling, juggling, and penalty shots for 133 takes across 12 unique participants. 7 of these participants were experts with 10+ years of experience, whereas the remaining 5 participants were casual soccer players. All videos were collected at a university soccer field.

Our study protocol was approved by the Institutional Review Board (IRB). All participants signed a consent form before participating in the study.

10.K. University of Pennsylvania

The University of Pennsylvania focused on capturing videos of experts of various levels playing musical instruments, dancing, and cooking. Over the spring and summer of 2023, UPenn captured 521 usable takes across 95 participants for the consortium's collections with up to 7 views.

One primary goal of this project is to capture detailed body movement, especially hands, across ego view and exo views. We work to ensure highly engaged experts enjoy demonstrating their full skill capability.

The hand-object/instrument interaction region is the key to understanding human activities and evaluating their skills. Comprehensive hand pose information is especially important for the full analyses of scenarios collected at UPenn, especially the music scenario, where slight differences in finger motion result in entirely different performances.

We also observed that experts had a tendency to not need to look at their hands during play. Thus, we found the initial data capture using the general camera setup to lack crucial visual information in such scenarios due to:
(1) (in ego view) limited field of view of Aria glasses, and skilled experts don't need to look at their hands,
(2) (in exo views) frequent occlusion and self-occlusion caused by participants' motion.

We added two cameras to maximize the view coverage.
Head-mounted Camera: The head-mounted camera on a helmet angled downwards to capture the hand/body region: (1) (ego) it follows the subject's body motion faithfully, and (2) (exo) it is designed to focus on the hand-object interaction region with much less self-occlusion. Empirically, we found this additional camera is crucial for capturing guitar, violin, and cooking scenarios.
Overhead Camera: We replace the head-mounted camera with an overhead camera in (1) piano scenarios, where the overhead camera can have similar performance, and (2) dance scenarios, where the helmet can dramatically worsen the experience and performance of the participants.

We believe the goal is not to maximize the number of hours captured but to have the participants show (1) diverse techniques to build models for the scenarios, and (2) unique techniques to demonstrate their skill levels.

To get the most representative recordings of the participants, we aim to maximize their engagement during the data capture. Specifically, we (1) walk through the whole process with the participants before the data capture to familiarize them with the setup (2) let them choose their favourite music piece to play or dance with in music and dance scenarios; and (3) have a narrate and act section for the musicians to demonstrate how they feel about their performance.

Music For musical instrument playing, classified as a "physical" activity, we captured takes of musicians (1) warming up (scales and or Etudes) (2) sight-reading simple sheet akin to Suzuki Practice books or Etudes exercises, and (3) freeplaying. We captured takes of violin, piano, and guitar, with a duet between a cello and a violin for one trial. Participants were recruited from a diverse pool of musicians, spanning the Penn Orchestra, local music schools, and independent music students. This pool's experience ranged from professional instructors and performers to complete newbies. We totaled 275 takes over 37 participants. Notably, during the shooting, we observed that participants were particularly uncomfortable with the helmet used to mount the GoPro; it interfered with their head movements and the bow sometimes ended up knocking against the mounted GoPro. To combat this, we added additional cushioning to depending on the subject's head shape and broke sessions into chunks to allow for breaks.

Cooking Cooking, categorized as "procedural", consisted of preparing four dishes: an egg dish, a salad, a noodle dish, and a dessert. The group of participants consisted primarily of Penn students with experience ranging from amateurs to hobbyists. Professionals were unavailable due to scheduling conflicts. We totaled 81 takes over 20 participants. The entire filming process was undertaken within a three-week span, primarily at the apartment of one of the team's participants. This location expedited our data collection for this task by providing a stove and fridge for regular use.

Dancing Dance captures, classified as a "physical", consist of four takes of dancers performing dance routines to a song. The dance types recorded included Lindy-Hop Jazz, Bollywood, Latin, and Chinese Folk Dance; across these genres, we totaled 165 takes over 38 participants. The Lindy-Hop Jazz dancers came from the Jazz Swing Attacks, a dance club in Philadelphia. Contact was established via Instagram, and data, collected weekly over a month. This group contained a balanced mix of experienced instructors and beginner dancers. The Bollywood dancers, the Drexel's Philly Maza, were recorded in the Drexel Engineering Building. They compete nationally but routinely train beginner recruits. The Chinese Folk Dancers were

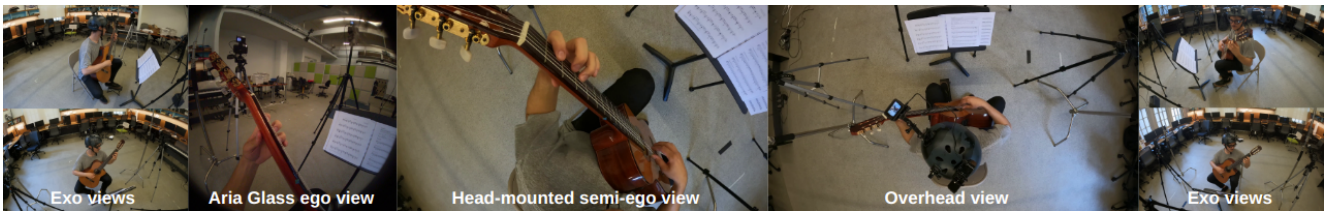


Figure 19. The Aria Glass ego view, head-mounted semi-ego view, overhead view and other static exo views in playing guitar in Philadelphia, PA, USA.

members of the local Great Wall Chinese School's dance club and independent student volunteers with prior competitive dance training. These were captured in the SIG Lab for collections.

All participants were confirmed to be at least 18 years of age by the time of participation and gave written consent for participating in these data collection trials. The consent form, in compliance with IRB guidelines, but gives participants the choice to back out. The collected information on basic demographic information should not be used to identify participants individually. All other data collected per participant (prior experience with task, average times spent per session, etc) could not be used to identify participants.

10.L. University of Tokyo

In Tokyo, we collected video data for three scenarios: cooking and health for procedural activities and soccer for physical activities. We followed the standard camera configuration and calibration procedure of the Ego-Exo4D dataset for all scenarios. In the following paragraphs, we will describe the specifics of each scenario, particularly the unique aspects of our data gathering.

Cooking We recruited 12 Japanese participants living in the Tokyo area through a temporary employment agency. The gender and age of the participants were balanced to collect diverse behavior patterns. The participants cook three days or more each week in their daily lives. Each participant prepared three dishes: an omelet, a white radish & lettuce & tomato & cucumber salad, and a sushi roll. We recorded both versions with and without narrations for each dish and participant. A one-page summary of each recipe was provided before data collection and was shown during video recording so that the participants could prepare the dishes smoothly, and the procedure of each recipe should be consistent between the participants.

All video recording of the cooking scenario was done in a rental kitchen studio equipped with an island kitchen and all the necessary kitchenware for four consecutive days. The studio is situated in a busy location in downtown Tokyo, and some external noises, like ambulance sirens, were audible during the recordings. We collected 68 takes from 12 participants. Of the 68 takes, 34 takes are with narrations, and 34 takes are without narrations. The length of each participant's making each dish twice with and without narrations is about 35 minutes, ranging from 24 min 27 sec (Sushi roll) to 55 min 4 sec (Salad). The length includes time for the camera synchronization procedure.

During the recording of the cooking scenario, we discovered a flickering issue in some of the video data due to the incompatibility between the Aria Glass sampling rate and the power frequency in Tokyo. To overcome this issue, we attempted to shoot as much as possible in daylight and adjusted the fps when using artificial lighting. While processing the videos, we discovered some exo videos had decoding errors due to damaged frames. Each corrupted video contained one to three damaged frames for an unknown reason. To address this, we re-encoded these videos by replacing the damaged frames preventing decoding with the nearest good frame. Note that some videos still contain damaged frames as long as those frames did not influence decoding. In addition, the original MP4 files recorded by GoPro contain 4 streams: video, audio, data0, and data1, but the re-encoded videos only contain a video stream and an audio stream.

Health We recruited 17 Japanese participants living around Tokyo, Japan, through a temporary employment agency. The gender and age of the participants were balanced to collect diverse behavior patterns. We recorded videos of the 17 participants performing two tasks: COVID-19 rapid antigen testing using three test kits and performing CPR on a mannequin. We conducted all recordings in the same meeting room on campus over two days. For the COVID test, an instruction manual of each test kit was provided to the participants before and during the recording. Also, we did not show the participants or record any COVID test results for privacy protection. For CPR, the participants took an introductory lifesaving course provided by the Tokyo Fire Department before recording. Besides, a one-page summary of the CPR procedure was provided before the data collection and shown during the video recording. This is so that the participants can perform CPR smoothly and the procedure is consistent among the participants. We collected 73 takes from 17 participants. All of the CPR procedures (17 takes) were recorded without narration. For the

COVID test, we recorded the videos of the 17 participants using the three test kits (51 takes) without narrations. The video length of each participant's performing CPR is 9 min 48 sec on average, including the camera synchronization procedure. Similarly, the video length of each participant's using the three COVID test kits is 29 min 22 sec on average. Additionally, we recorded extra takes of 5 participants out of the 17 using a test kit with narrations (5 takes in total).

Soccer We gathered videos of 14 Japanese participants, each performing three fundamental soccer drills: dribbling and juggling for two minutes each and penalty kicks ten times. Of the 14 participants, 13 are soccer players from a university football club. We recruited them through the staff of the club. The remaining one participant is not from the club but is an expert with over ten years of soccer experience. All the participants are male, and their age ranges from 18 to 30s. We recorded the videos on an outdoor soccer field at a local university over four days, with three to four participants participating each day. For juggling, we instructed the participants to include various movements such as juggling with thigh, inside and outside of feet, and alternating feet. For penalty kicks, we instructed them to shoot to the right side of the goal 5 times and to the left side five times. During penalty kicks, a helper aids the participant in retrieving the ball. This helper stands within the goal area and might be recorded by some cameras. We collected 42 takes from 14 participants. All the takes were recorded without narrations.

Our university's institutional review board reviewed and approved our study protocol. We explained the objective and the range of use of the videos through documents and took consent from each participant before the recording. In particular, we took the consent not to blur their faces to keep the naturalness of the videos.

Contributions Statement

Los Andes University Pablo Arbeláez - lead coordinator for data collection and collaborator on the overall project design; Maria Escobar - data collection for all phases, design of the collection setup and workflow, data inspection, ingestion, encoding, and metadata generation; Cristhian Forigua - data collection for all phases, participant recruitment, consent forms design, data inspection, communication with recording sites; Cristina González - data collection for phase 1, design of the collection setup and workflow, IRB management; Angela Castillo - data collection for phase 2, manual data inspection, and data analysis.

Georgia Tech James M. Rehg - lead coordinator for data collection and protocol design, and overall project manager; Bikram Boote - lead coordinator for data collection, including recruiting and ingestion; Fiona Ryan - contributed to data collection; Audrey Southerland - lead coordinator for IRB development, contributed to recruiting.

National University Singapore Mike Zheng Shou - lead coordinator for data collection and protocol design, and overall project manager; Joya Chen - contributed to protocol design, camera setup design, data collection for all phases; Jia-Wei Liu - contributed to protocol design, camera setup design, data collection for all phases; Xinzhu Fu - contributed to data collection for all phases; Chenan Song - contributed to data collection for all phases.

Meta Andrew Westbury was the lead for data collection at our site, selecting scenarios, organizing capture sessions, recruiting participants, organizing and transferring data, and obtaining required approvals. In California, Hao Tang and Kevin Liang also supported all these functions, focused on bike repair. In New York, Devansh Kukreja and Alex Dinh lead collection for cooking scenarios. Miguel Martin also supported California-based collections and organized our local camera rig. Chefs Eton Chan and Dominic Ainza supported all culinary collections with technical guidance, recruitment, and coordination. Dimitri Elston coordinated and was the technical lead on bike collections. Adrian Salas supported pilot bouldering collections in California. Across all Ego-Exo4D collections, Devansh Kukreja continuously communicated and refined the recording procedure with universities, and problem-solved local recording issues.

University of North Carolina at Chapel Hill Gedas Bertasius - lead coordinator for data collection; Md Mohaiminul Islam - the main contributor to data collection and metadata processing across all scenarios; Wei Shan - contributed to data collection and metadata processing for the music and soccer scenarios; Jeff Zhuo - contributed to data collection and metadata processing for the soccer scenarios; Oluwatumininu Oguntola - contributed to participant recruiting and data collection for the music scenario.

Carnegie Mellon University Rawal Khirodkar developed the automatic 3D body keypoints extraction pipeline and collected a subset of the soccer, bike mechanic, and cooking sequences for the CMU portion of the dataset. Sean Crane was in charge of the data collection, IRB documents, capturing data, working with participants and processing the data for CMU. Abrham Gebreselasie ran the actionformer-based baseline for demonstration proficiency benchmark. Eugene Byrne served as the engineering lead for the initial design and implementation of the dataset, camera rig, processing pipeline and keystep annotations while at Meta. Subsequently at CMU, he assisted in the recognition benchmarks, implemented Ego-Exo transfer [82] (1 of the 3 baseline methods for keystep recognition) and the initial implementation of keystep action detection [189], and assisted in annotation/data quality generally.

Simon Fraser University Sanjay Haresh was the lead coordinator for data collection, including recruiting, data in-

gestion, and data analysis. Yongsen Mao also contributed to the data collection pipeline, recruiting, data ingestion, metadata annotations, and statistics computations. Manolis Savva advised on data collection, protocol design, and overall project management. We acknowledge the assistance of Hanxiao Jiang and Armin Kavian with recruitment and data collection.

University of Pennsylvania Edward Zhang led data collection efforts at UPenn and played a key role in subject recruitment, subject information collection, and on-site data recording. Jinxu Zhang led data management, information logging, and data transfer. Shan Su is the overall project lead, focusing on determining good camera configurations based on 3D reconstruction feasibility and fixing issues in data post-processing of time synchronization and take separation.

University of Tokyo Yoichi Sato served as the primary coordinator for data collection, while Ryosuke Furuta was responsible for data collection across all three scenarios, participant recruitment, and IRB submission. Zecheng Yu and Masatoshi Tateno provided support for data collection and were responsible for managing and transferring data. Takuma Yagi helped with the IRB submission process.

Indiana University At Indiana University, Weslie Khoo led the IRB protocol design and compliance, arranged logistics such as ordering equipment, and designed and oversaw the cooking scenario data collection. Yuchen Wang and Ziwei Zhao co-led the participant recruitment and data collection for all three scenarios. Ziwei Zhao led data preparation and transfer. We also acknowledge Manasi Swaminathan who assisted with data collection and video synchronization.

IIIT-Hyderabad Avijit Dasgupta was the lead on the ground for data collection in Hyderabad helping in organizing capture sessions, data collection, and managing and transferring data. Siddhant Bansal helped in the early stages with IRB application, consent forms, and pilot studies. C. V. Jawahar was the lead coordinator for data collection helped in selecting the scenarios, and recruiting the participants.

University of Minnesota Hyun Soo Park oversaw the overall effort at the University of Minnesota, Twin Cities, including protocol design and data collection. Zachary Chavis led the IRB protocol design and compliance, arranged logistics such as ordering equipment, participant recruitment, and designed and oversaw all scenarios of data collection. Anush Kumar assisted data collection for all scenarios.

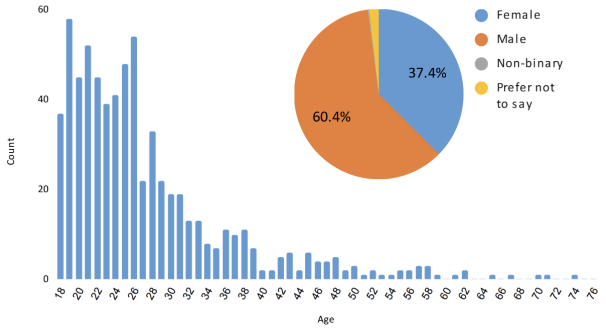


Figure 20. Participants' self-reported demographic information

11. Participants

Next we overview the background of the participants (camera wearers) in the Ego-Exo4D dataset.

11.A. Demographics

We provide self-declared information on ethnic groups by the participants. Sharing this information was optional for all research subjects. Ethnicity is reported based on location specific categories as defined by the relevant partner lab. No such information was gathered from research subjects participating in our collections in California, New York, and Pittsburgh, Pennsylvania. Aggregated gender and age information of all participants in Ego-Exo4D is provided in Figure 20.

Atlanta, Georgia, USA 100% of participants that reside in Fulton County, Georgia self-reported their ethnic group membership as follows:

Ethnicity	Number of participants
Asian	23
White	8
Hispanic/Latino	3

Bloomington, Indiana, USA 100% of participants that reside in Monroe County, Indiana self-reported their ethnic group membership as follows:

Ethnicity	Number of participants
Asian	22
Black	1
Middle Eastern	1
White	18
Prefer not to say	1

Minneapolis, Minnesota, USA 100% of participants that reside in Hennepin County, Minnesota self-reported their ethnic group membership as follows:

Ethnicity	Number of participants
White	41
Hispanic/Latino	4
Asian	8
Black	1

Tokyo, Japan 100% of participants that reside in Tokyo self-reported their ethnic group membership as follows:

Ethnicity	Number of participants
Asian (Japanese)	45

Hyderabad, India 100% of participants that reside in Hyderabad self-reported their ethnic group membership as follows:

Ethnicity	Number of participants
Asian (Indian)	95

Chapel Hill, North Carolina, USA 100% of participants that reside in Orange County, North Carolina self-reported their ethnic group membership as follows:

Ethnicity	Number of participants
White	20
Indian	1
Asian	13
African American	9
Hispanic/Latino	3
Prefer not to say	3

Vancouver, British Columbia, Canada 100% of participants that reside in Vancouver self-reported their ethnic group membership as follows. Please note that research subjects in this case opted not to use any assigned category and independently described their identity.

Ethnicity	Number of participants
African/Nigerian	4
Asian	9
White/Caucasian	10
Chinese	26
European	1
Iranian/Persian	14
Italian	1
Jamaican	2
Kazakh	1
Kyrgyz	2
Middle Eastern	1
Mixed	3
South Asian	2

Philadelphia, Pennsylvania, USA 100% of participants that reside in Philadelphia Country, Pennsylvania self-reported their ethnic group membership as follows:

Ethnicity	Number of participants
White/Caucasian	10
Asian	30
African American	3
Hispanic/Latino	4
Prefer not to say	43

Singapore 100% of participants that reside in Singapore self-reported their ethnic group membership as follows:

Ethnicity	Number of participants
Chinese	65
Indian	3
Singaporean	2
Indian/Chinese	2
Prefer not to say	17

Bogota, Colombia 100% of participants that reside in Singapore self-reported their ethnic group membership as follows:

Ethnicity	Number of participants
Black/ Afro-descendant/ Afro-Colombian	7
Mixed	104
Palenquero	1
Raizal	1
White/Caucasian	38
Prefer not to say	23

11.B. Participant Surveys

To ensure consistent, high quality annotations (which will be discussed in Appendix 13), we identified three aspects from data collection where additional information from the participants' recording is crucial. These are: the *skill* of the participant, the *objects* they are using, and the *actions* they are completing. The three aspects represent information that either cannot be captured by other annotators after the fact or could result in noise within the annotation process. Accordingly, we designed participant surveys to cover key information regarding the participants' *skill*. To capture the *actions* and *objects* with which they interact, we ask participants to perform a round of first-person narrations called "narrate-and-act" (cf. Appendix 12.B).

Participant surveys were separated into two: a pre-task questionnaire and a post-task questionnaire. The pre-task questionnaire aims to capture the participant's perceived skill level whereas the post-task questionnaire captures the participant's reflection on how well the task went. The list of questions for both questionnaires can be found in Table 3 with questions/answers designed for consistency and ease of filling in, as participants would be filling these out before/after each recording. This involved using multiple choice and Yes/No answers with open text fields being utilized sparingly.

Pre-task Questionnaire Within this questionnaire, the aim is to capture the participant's perceived skill level regarding the task that they are about to perform. To maximize consistency across participants, we forgo asking participants to self-rate themselves as experts, novice, etc.—instead, we ask participants more quantifiable questions such as length of time the participant has been doing this task and frequency.[5] Additional questions regarding whether the participant has taught the task to others and whether they have recorded/watched videos about the task also give signals regarding the proficiency of the user. Note that the information gathered is scenario- and take-specific, since the number of hours to be considered an expert or amateur can be wildly different (e.g. cooking vs. bouldering).

Post-task Questionnaire The post-task questionnaire focuses on capturing details of the take and how it went according to the participant. Questions are asked about mistakes, objects, and time taken compared to predicted time, which is beneficial for downstream annotation.

The survey data will be shared with the dataset.

[5] We also obtain proficiency ratings for the participants via our expert commentators (cf. Appendix 12.A).

	Question	Answer Type
Pre-Task	Recording Location	multiple choice
	How many times do you estimate you have done this task?	multiple choice
	How often do you carry out this task?	multiple choice
	How many years have you been doing this task?	multiple choice
	Have you taught this activity to others before?	Yes/No
	Have you recorded a video of yourself carrying out or explaining this task before?	Yes/No
	Have you watched videos of others doing this task before?	Yes/No
	Do you have any qualifications/professional training that are related to the task?	Yes/No
	How long does it typically take you to complete this task?*	text
	How long would you typically spend in one practice session of this task?†	text
Post-Task	Self Reported Quality	multiple choice
	Completed Route?‡	Yes/No
	What mistakes/errors did you make during this task?	text
	Any issues with the familiarity of the tools/location?	text
	Did it take longer/shorter than your initial expectation and why?	text
	How did you find wearing the camera?	multiple choice
	How easy was the setup for recording?	multiple choice
	Any other comments to take on board?	text

Table 3. Questions for the pre-task and post-task questionnaires. *: Only applicable for non-dance/non-music scenarios. †: Only applicable for Dance/Music scenarios. ‡: Bouldering scenario only.

Figure 21. Examples of the three different annotation styles: narrate and act, atomic action descriptions, and expert commentaries from four of the scenarios (bike repair, health, dance, and basketball). We also include word clouds which highlight the differences in vocabularies per scenario. In narrate and act text, we see how the participants briefly describes what they are doing and why, whereas the atomic action descriptions provide strictly a statement about the visible actions. The expert commentary offers an expert's critique of what is shown, commenting on strengths and weaknesses and explaining how the participant's actions affect their performance.

12. Language Descriptions

As introduced in the main paper, Ego-Exo4D provides three forms of parallel text corpora for the video: expert commentary, narrate and act, and atomic action descriptions. Figure 21 and Table 4 show examples from different scenarios highlighting their distinctions in style and point of view. In the following we elaborate on their protocols, show examples, and analyze their differences.

Ego-Exo4D's language annotations are not targeted at any particular benchmark task or taxonomy. They aim to be a general resource that will inspire new language-vision possibilities, such as learning how to generate coaching or teaching advice given a demonstration video, learning how to spot skill and lack of skill, or learning general multimodal embeddings in the procedural and skilled activity domain. We also anticipate the temporally grounded descriptions to be valuable for pre-training foundation models [85, 123] or automated video captioning [59, 111, 199]. Furthermore, the time-anchored aspect of the three language annotations provides the opportunity to retrieve time points in specific Ego-Exo4D videos that correspond to queried moments, actions, or phrases. Finally, they are valuable to mine the dataset for the distribution of objects and activities present, e.g., for taxonomy formation.

12.A. Expert Commentary

Ego-Exo4D offers a new video-language resource specifically designed to support the study of human skill. We mobilize 52 expert commentators, each with rich qualifications and skills in one of our focus domains, to provide time-anchored critiques and explanations of all the skill demonstrations in the dataset. Specifically, our expert commentators use spoken word to describe what is most effective or ineffective about the camera wearer's actions, review the quality of the execution, and identify mistakes. Experts also provide an overall proficiency rating on each skills demonstration, assessing how well the task was performed with a short written justification. The following section describes the qualifications and background of our experts; commentary instructions and tooling; and statistics on collected commentaries to date.

Experts and Expert Qualifications Over a period of three months, we recruited experts from across each of Ego-Exo4D's eight core domains, working with a selection panel of ten individuals to review qualifications and engage in more than 68 interviews. Our selection criteria focused on technical skills, communication, and task completion along with performance reviews during a live video commentating exercise. Among the numerous candidates we interviewed, we were pleased to select 52 to join the effort.

On average, 90% of recruited experts possess more than 10 years of professional experience and all have served during this time in the capacity of a coach, instructor or mentor. All experts further have either an advanced degree in their domain of focus or an industry certification. Certi-

Proficiency score: 10
Commentary: Great footwork. He's using dribble to set up his footwork and his shot. Stepping onto that left foot bringing the ball. I love that his eyes and head are up. He already knows where he's going to go.

Proficiency score: 5
Commentary: The dancer's hands should be a bit higher. This line should be completely straight in front of him at his shoulder length. It shouldn't be beginning to dip lower.

Figure 22. Two examples of expert commentary and proficiency scores, along with spatial drawings (red) done by the expert to augment the spoken comments.

fication authorities include the US Soccer Federation, the American Culinary Federation, USA Climbing, the American Red Cross, Trek Bikes, and New York State's Initial Certification in Teaching Dance, among others.

Multiple individuals were recruited across each domain, with the goal of generating language and expertise diversity. Table 5 shows the number of expert contributors per domain. Due to employment considerations, all experts are residents of the United States.

Commentary Instructions and Tooling Experts are provided with two time-synchronized videos of each Ego-Exo4D skills demonstration—one showing the egocentric view and another providing a single exocentric perspective specifically selected by annotators as the view that provides the best visibility on the scene. Experts are then asked to watch the video in full without commenting to gain an understanding of the skills demonstration and plan out important points to note in their commentary.

After this initial viewing, experts are asked to provide a numeric proficiency score on the scale of 1 (least skilled) to 10 (most skilled), rating how well the camera wearer performed the task. A short written description of why the chosen score was selected is also provided at this stage. In many cases, experts coordinated within their domain group

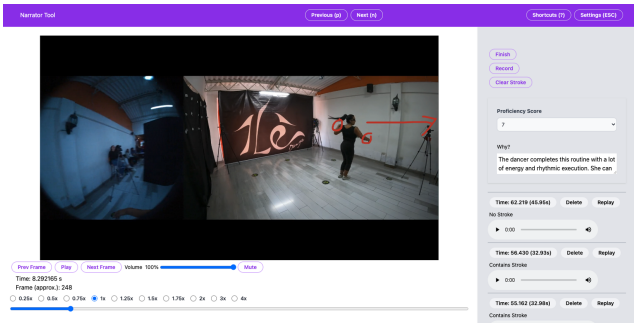


Figure 23. Our expert commentary web tool called *Narrator* provides an easy-to-use platform for experts. Experts can stream video, record audio commentaries, and provide proficiency ratings and justifications. The tool also supports drawing on the video feed (see red arrow and circles on the right frame), allowing for manual spatial grounding during commentary.

to calibrate this scoring. While each proficiency rating is at the discretion of individual experts, domain-level calibration provided a general framework for assessing skills demonstrations.

With proficiency scores noted, experts play the video from the start and pause at critical moments of the demonstration to record verbal descriptions of what is observed to be most effective or ineffective about the camera wearer's actions, offer performance tips, identify mistakes or suboptimal execution, and why they matter. In all cases, commentary is time-anchored and retrospective, focusing on insights and perspectives relating to actions visible to that point in the video. We choose to collect commentary as verbal recordings in order to maintain the naturalness of the performance descriptions and do so quickly. In all cases, experts had the option to use a "telestrator" tool to enhance their commentary with freehand sketches to spatially localize information or otherwise help explain a point. See Figure 22. We transcribe the commentaries automatically with OpenAI's Whisper for automatic speech recognition [125]. Table 4 (right column) and Figure 21 shows example commentaries. In total, our experts have dedicated more than 6,000 hours of effort to date to provide this commentary.

To collect expert commentary, we developed a web-based tool, which is open sourced as part of the Ego-Exo4D dataset and benchmark suite. Known as the Narrator, this application supports video playback for Ego-Exo4D skills demonstrations, records time-stamped verbal commentary, and allows exporting and viewing commented videos. As a web-based platform, the Narrator can be simply accessed through a browser, with minimal set-up and less restrictive system requirements compared to tools requiring local installation. These attributes made it efficient to onboard and manage our geographically distributed experts. We acknowledge the EPIC Narrator [27] as the open-sourced in-

Domain	Atomic Action Description	Narrate and Act	Example commentary
Cooking	C turns on heat on the gas burner.	So I'm going to start out by boiling some water.	Here the preparer is checking the pasta for done-ness. It's important to do this and not rely on what a package says. Use a package that gives you cooking time as a guideline and start to check your pasta, you know, a few minutes before the maximum amount of time given for cooking that specific pasta.
Health	C inserts the nasal swab in the buffer test tube on the covid test kit pack with his right hand.	Open the newly picked up tube, place the swab in the tube, stirring the swab in the tube.	So this individual has done a great job of making sure that her nasal passageways have adequate time in contact with her nasal swab. Something that might make it a little bit easier for her is if she could tilt her head back just a bit so that she wouldn't have to strain quite so much to get that access. Additionally, she did a great job making sure that the nasal swab was about an inch into her nose.
Bike repair	C holds the bike wheel with her left hand.	And then I will locate the location of the valve cap and pull the tube out of the wheel.	It's a great method to always double check or do a pre-check before beginning work on a bicycle to make sure the issue that you are working to fix is the only issue that is occurring. If not, you could find a secondary issue or something else that may be greater than the one you are currently working on.
Music	C puts the bow on the violin with his right hand.	So regardless of how tricky left hand passage work is you want to always keep your bow completely independent.	This is a really great use of the bow and decision to play in this middle third of the bow. This is exactly where they should be playing. And we can hear that the note envelope is very consistent and that it's very controlled and that it also allows the rhythm to be stable...
Basketball	C runs towards the hoop with the basketball.	Now I'm going to do a reverse layup, stepping right, left, going up with the right hand.	As the ball goes through the basket, she catches the ball and does an excellent job of keeping the ball high, never allowing the ball to drop down to her waist area, but keeping the ball high in her upper chest, neck area throughout the drill...
Bouldering	C places both hands on a red hand hold.	So I know that a lot of these holds, I'm going to need my weight leaning to the left to utilize	Once the climber recovered from the foot cut, the climber pasted the right foot on this foot jib and then did a toe match. So brought this foot in and then dropped the right foot down and to the right to again counterbalance so that the climber can then move their left hand out left. But at this point the climber is just a little too gassed to be able to make this move, which is unfortunate.
Soccer	C kicks the ball to the right with his right foot.		Angle approach, start position is good, maybe slightly squarer than 45, but again because the intended outcome from previous actions is into the left, by being a little squarer is going to help him be able to rotate his hips to move to the left, but on a slight angle is good and help him with his technical action.
Dance	C moves her right leg forward while swinging both hands.	Ring and wing, one, two, one, two, three	She is doing these steps in place when she's traveling forward. At this point, she really could be further forward all the way, still on the screen, but towards the edge of the screen, if she was to take bigger steps. And she could take bigger steps if she bends her knees and lowers her center of gravity and then extends her leg outward...

Table 4. Example excerpts from all three language types. Experts are charged with critiquing the performance of the participants, pointing out strengths and weaknesses and explaining how the participant's approach influences the quality of their skill demonstration. Narrate and act focuses more on what the camera wearer is doing and, sometimes, briefly why. Atomic action descriptions are about the specific actions seen.

Domain	# experts
Basketball	6
Music	6
Soccer	5
Dance	6
Bouldering	6
Health	5
Cooking	11
Bike Repair	7

Table 5. Number of experts providing commentary in each domain

spiration and source code for this initiative.

Commentary Analysis We ask at least 2 and as many as 5 distinct experts to commentate any given video, to provide a variety of language and opinions. Altogether, we have collected 117,812 pieces of commentary. On average there are 7.5 pieces of commentary per minute of video, and typically the expert gives about 4 sentences of commentary every time they pause to react to the video. Table 5 shows the number of skills demonstrations with expert commentary per domain. Overall, we believe the commentary is a unique window into the skilled actions that (through language) surfaces many subtleties about the actions not evident to the untrained eye.

12.B. Narrate and Act

The participant surveys (Appendix 11) capture the *skill* of the participants, but they do not provide first-hand information regarding the *actions* being completed and *objects* they used. We capture these using "Narrate and Act": a separate take recorded with the participant in which they narrate what they are doing and how they are doing it.

Participants were asked to complete the Narrate and Act take as if they are teaching someone else as part of an instructional video or how-to guide. In this way they would talk about what they were doing as they carried out an action. This is akin to the kind of narrations done by people creating how-to videos, though with less stylization and without any professional post-production editing. It is interesting to note that some participants did this naturally requiring little prompting. For some activities which were more intense, such as dancing or bouldering, we asked participants to instead narrate either just before or just after the action to reduce the difficulty of doing this live.

Compared to the third-person expert commentary (above), these narrations are first-person and delivered at the same time of the actions. Generally the commentary is richer in constructive feedback about the quality of the activity, whereas the narrate-and-act narrations are interesting for their simultaneous nature and first-person analysis of what the participant is doing. The behavior in this extra take is expected to differ from that of the non-narrated tasks, in that it is likely that the participant will complete the scenario more slowly than normal to concentrate on explaining what they were doing. However, the narrate and act takes provided benefits in reducing annotation noise and they can potentially be used for multimodal learning as is currently explored in the literature with how-to video narrations [9, 89, 103, 104]. Table 4 (third column) and Figure 21 show examples.

12.C. Atomic Action Descriptions

Inspired by the "narrations" in Ego4D [47], our third language resource provides atomic action descriptions of the camera wearer and his or her environment, when relevant, as captured by the Ego-Exo4D videos. Compared to the expert commentaries and narrate-and-act narrations—which both emphasize the "why" and "how"—these atomic action descriptions are focused on the objective "what" of what is happening when, written in free-form text from the perspective of a third-party observer.

Annotation description We present each take to the annotators as a collaged video consisting of the Aria egocentric view, left and right grayscale SLAM, four or five fixed-position exocentric cameras, and single-track composite audio; for a subset of videos, a helmet-mounted GoPro view is also available. Annotators were asked to provide a play-by-play description of what happens, as seen across any of the views. Potential contents include actions by the Aria camera wearer, other individuals interacting with the camera wearer, and relevant environmental events. Each narration is atomic and time-anchored: as much as possible, each narration should roughly be limited to one verb and have a single associated timestamp, roughly within a second of its occurrence in the video. For consistency across narrations, the Aria camera wearer in each take is referred to as "C" in all narrations (*e.g.* "C picks up a wrench."). Other individuals are referred to by other letters (*e.g.* "Man X kicks the soccer ball back to C."); these letter labels are not necessarily consistent across takes, but refer to the same individual within a take. Many videos are narrated by two independent human annotators, and we make each narration set separately available. Table 4 (second column) and Figure 21 show examples. See Table 6 for atomic action descriptions summary statistics.

Visibility Because of the multi-view nature of the Ego-Exo4D capture rig, certain actions or events may not be visible across all camera feeds. While we hope Ego-Exo4D leads to increased attention toward multi-view learning, many existing systems fundamentally assume a single view

Category	1x Coverage	2x Coverage	# of Descriptions	Descriptions Per Minute	Unique Nouns	Unique Verbs
Basketball	778	116	50299	53.330 (+- 26.049)	201	134
Bike Repair	202	160	31317	24.891 (+- 9.555)	642	393
Cooking	360	266	189225	27.745 (+- 12.843)	1744	823
Dance	307	417	43663	30.852 (+- 13.915)	504	468
Health	299	97	43769	24.304 (+- 11.234)	619	384
Music	85	75	10695	4.278 (+- 8.969)	255	163
Rock Climbing	1270	103	32246	32.350 (+- 11.974)	301	224
Soccer	225	53	31253	38.467 (+- 23.957)	229	125
All	3526	1287	432467	31.293 (+- 20.209)	2924	1481

Table 6. Atomic action descriptions per domain statistics.

at a time; if a camera does not have a view of the narrated action or event, this may lead to a confusing learning signal, or pose an impossible ask for a model to infer. Thus, we also ask that the annotators answer two additional question per narration: 1) an indicator of whether the narration is visible from the egocentric camera, and 2) which (if any) of the static exocentric cameras provide the best view. If there are multiple equally good views, annotators are free to pick any. In particular, we found this *best exocentric view* helpful for other Ego-Exo4D annotation efforts: the narration visibility tags played a role in exocentric view selection for both the correspondences benchmark (Section 13.A.1) and expert commentary (above), and frame selection for hand and body pose (Section 13.D).

Relation to other Ego-Exo4D annotations While bearing some similarities to our other annotation workstreams, these atomic action descriptions exhibit some key differences:

- *Keysteps* (Section 13.B.1): While some atomic descriptions are similar to certain keystep names, narrations are free-form, without intentionally imposing any shared vocabulary. Atomic descriptions also aim to capture all actions and events, regardless of relevance as immediate steps towards the scenario's goal, and cover all domains, not just procedural.
- *Expert Commentary* (Section 12.A): As atomic description annotators do not necessarily have the years of experience of the domain experts, the resulting text is often written from a layperson's perspective, and focus on what is happening as opposed to providing critiques, analysis, or explanation. Atomic descriptions also tend to be shorter and denser than expert commentary.
- *Narrate and Act* (Section 12.B): While also a language resource, narrate and act's descriptions are expressed from a first person perspective and can share intrinsic motivation behind actions. In contrast, atomic descriptions are written from the perspective of an outside observer.

12.D. Comparison of the Language Statistics

In Figure 24 we further emphasize the characteristics of each text corpus across three axes: total vocabulary size, average number of captions per video, and caption length. See caption for details.

Figure 25 shows word clouds per scenario and annotation type highlighting the differences in vocabulary and word frequency.

Contributions Statement

Kevin J Liang co-developed the expert commentary guidelines, interviewed and onboarded experts, helped test and suggest features for the narrator tool, and contributed to program management; he also developed the atomic action descriptions guidelines, helped coordinate annotations, and contributed to paper writing. Michael Wray contributed to the definition of the expert commentary guidelines; provided feedback for experts; co-developed the narrate-and-act guidelines and the pre-task/post-task questionnaires; and contributed to paper writing. Kristen Grauman proposed the expert commentary idea, co-developed the guidelines, interviewed and provided feedback to experts, and contributed to paper writing. Andrew Westbury implemented expert commentary, recruiting, mobilizing and managing our experts and workplan. Miguel Martin contributed to the atomic action descriptions annotation guidelines and produced the annotation files and associated tutorial code, and for expert commentary, he authored the initial version of the Narrator tool, transcribed the commentaries, and produced the annotation files and associated tutorial code.

Changan Chen contributed to the development of the narrator tool; provided feedback for experts; and contributed to paper writing. Siddhant Bansal contributed to the design of narrate-and-act, user questionnaires, object dictionaries, expert commentary cooking. Dima Damen proposed narrate-and-act data collection and user questionnaires and contributed to their design, and also contributed to expert commentary for cooking scenarios. Tiffany Davis provided significant program management support throughout expert commentary. Devansh Kukreja built the render flow to generate video collages for annotations.

Domain resource people from our consortium were

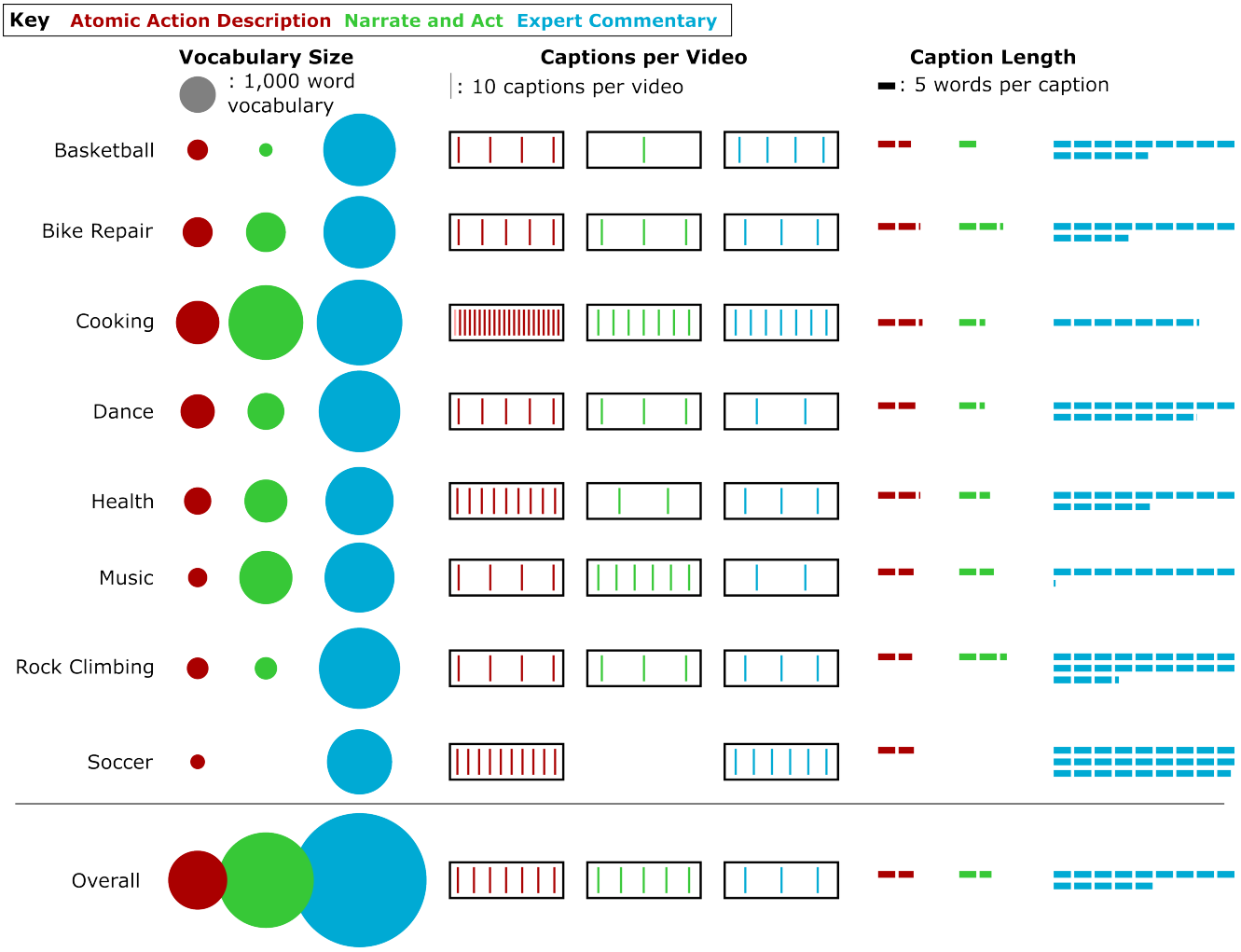


Figure 24. Comparisons between the vocabulary size (left) number of captions per video (center) and length of caption (right) for the atomic action description, narrate and act, and expert commentaries. Statistics are shown both per scenario and over the entire dataset. We see that the expert commentary tends to use a much larger vocabulary and more lengthy statements, since commentators are giving more elaborate statements of advice and explanation. The temporal density of the atomic action descriptions is greater than the other two forms, since the annotators are pausing to describe every single action of the camera wearer. Narrate-and-act comments use a vocabulary size in between the other two, reflecting the more free-form speech (compared to the written atomic actions) is used. Trends are mostly similar across scenarios, with the most noticeable differences being the temporal density; it is particularly high for both cooking and soccer. In the former, there are many procedural steps, whereas in the latter there are many instances of the drill being executed.

Dima Damen and Michael Wray [cooking], Kristen Grauman and Changan Chen [soccer], Gedas Bertasius [basketball], Kristen Grauman and Jianbo Shi [music], Andrew Westbury [health and bike repair], Kevin Liang [dance], Pablo Arbelaez and Maria Escobar [bouldering]

Ego-Exo4D's panel of expert commentators is: **Soccer**: John Bello, Phillip O'Kennedy, Lee Bakewell, Radcliffe McDougald, Thomas Harris **Music**: James Peterson, Trevor Minton, Andrea LaPlante, Ethan Fallis, Alex Rogers, Jacqueline Burd **Health**: Jasmine Higa, Angela Liszewski, Kristin Blanset, Melissa Robinson, Sonya Johnson **Dance**: Rolanda Williams, Deanna Martinez, Enya-Kalia Jordan, Rachel Repinz, Yauri Dalencour, Kathryn Hightower **Cooking**: Mark Manigault, Mary Drennen, Tiffany Davis, Reginald Howell, Rosanne Field, Donnie Murphy, Kiet Duong, Laura de Vera, Keegan Taylor **Bouldering**: Daniel Ramos, Mike Kimmel, Roy Quanstrom, Christopher Deal, Carmen Acuna, Kelsey Hanson **Bike repair**: Cesar Pineda, Walker Wilkson, Frank Trotter, Cordell Bushey, Dimitri Elston, Sam Arsenault, Aaron Hill **Basketball**: Elizabeth Blose, Raven Benton, Joseph McCarron, Cornelius Gilleyen, Cecil Brown. Aaron Jones

Figure 25. Word clouds for each scenario and annotation type. The vocabulary for atomic action descriptions typically focuses on the person's hands and how they complete the actions (e.g. using left/right/hand) whereas narrate and act describe the high level goals/objects. The expert commentary has the largest variety of words, including specialist words for each scenario such as swab/solution for health and axle/valve for bike repair.

13. Benchmarks: Annotations and Baselines

In this section we provide details on all the benchmark task definitions, their annotations, and our baseline models and results. We cover ego-exo relation (Sec. 13.A), followed by ego-exo recognition (Sec. 13.B), proficiency estimation (Sec. 13.C), and ego body and hand pose (Sec. 13.D).

Important: to ensure fair comparisons in any future work using Ego-Exo4D, researchers need to account for 1) the precise task input-output definitions and 2) the train/test/val splits available with v1 or v2 of the annotations. Specifically, for each task, when formally defining the inputs and outputs, we also explicitly specify which inputs are *excluded* from use, if any. Furthermore, there are two publicly released versions of Ego-Exo4D annotations: v1 is used to train/test baselines in this paper; the larger v2 will be used for future challenge leaderboards. Table 7 provides summary annotation statistics for all tasks of Ego-Exo4D. Again, these two points are important information for any future research done with Ego-Exo4D to ensure consistency of results in the literature to come.

13.A. Ego-Exo Relation

The family of ego-exo relation tasks deals with relating the video content across the extreme ego-exo viewpoint changes, in the form of either object-level matching (correspondence) or generation of one view from the other (translation).

13.A.1 Ego-exo correspondence

Annotations We annotate pairs of temporally synchronized egocentric and exocentric videos with segmentation masks for selected object instances from six scenarios: *Cooking*, *Bike Repair*, *Health*, *Music*, *Basketball* and *Soccer*. We exclude *Bouldering* and *Dance* from this benchmark as they have limited diversity of objects. We focus on objects used by the camera-wearer at any point during the execution of the activity and that are visible in both views for at least some frames of the sequence. These masks allow us to define object-level correspondence between the views.

We used a multi-stage annotation process for annotating paired ego-exo videos:

- *Stage 0: Object Enumeration*. Annotator marks each object that is active at some point of the egocentric video with a bounding box in a frame where it is clearly visible and provides a free-form textual description.
- *Stage 1: Egocentric video annotation*. Annotator watches the egocentric video and is also shown (a) text and (b) a bounding box for one of the objects annotated in the previous stage. Annotator then marks a segmentation mask for that object in all the video frames where the object is visible. Segment Anything [70] is leveraged to generate segmentation masks efficiently using only point clicks.
- *Stage 2: Exocentric video annotation*. As shown in Figure 26, the annotator watches a temporally synchronized exocentric video and is also provided with the (a) text and (b) several ego segmentation masks of this object. Annotator then marks a segmentation mask for this object

Benchmark	Annotation Type	EgoExo4D v1		EgoExo4D v2	
		Num Takes	Annotations	Num Takes	Annotations
Relations	Manual	1028	3419 objects 426K ego masks 611K exo masks	1335	5566 objects 742K ego masks 1.1M exo masks
Keystep recognition	Manual	1088	17 activities, 664 keysteps 27.6K ego segments (87h) 143K ego+exo segments (454h)	1088	17 activities, 664 keysteps 27.6K ego segments (87h) 143K ego+exo segments (454h)
Procedure understanding	Manual	374	6 activities, 186 keysteps 5.4K segments (18h)	374	6 activities, 186 keysteps 5.4K segments (18h)
Proficiency estimation	Semi-automatic	2987	2987 proficiency scores (demonstrator)	2987	2987 proficiency scores (demonstrator)
	Manual	912	19K "good" segments 20K "tips" segments (demonstration)	912	19K "good" segments 20K "tips" segments (demonstration)
Ego pose (Body)	Automatic	610	1.4M 3D / 5.7M 2D	2559	9.2M 3D / 46.87M 2D
	Manual	511	200K 3D / 1.03M 2D	1358	376K 3D / 2M 2D
Ego pose (Hand)	Automatic	749	2.1M 3D / 10.5M 2D	976	4.3M 3D / 21M 2D
	Manual	352	51K 3D / 285K 2D	458	68K 3D / 340K 2D

Table 7. Summary of annotation statistics for the different benchmark tasks of Ego-Exo4D.

Figure 26. **Multi-stage annotation process for Ego-Exo Relation annotations.** After enumerating all active objects in the egocentric video, an object is selected and annotated with segmentation masks in all frames of the egocentric video. Then, annotators are given the exo video as well as the textual descriptions and sample egocentric segmentation masks for the object of the interest, and mark segmentation masks for the specified object of interest in all the frames where it is visible.

Scenario	# Takes	# Objects	# Ego Masks	# Exo Masks
basketball	394	602	21820	31165
bike	210	714	53886	71763
cooking	478	3481	549507	888384
health	127	570	77596	86585
music	112	153	33624	5599
soccer	12	22	2411	2475
Total	1335	5566	741965	1091135

Table 8. **Relation annotation statistics.** We show statistics for each scenario including the number of takes, total number of objects annotated and the number of egocentric and exocentric segmentation masks.

in all the exo video frames, whenever the object is visible.

What are the objects of interest? We focus on objects that are *active* at some point during the execution of the activity. These objects are not only interesting because they are essential to the activity, but they are also challenging to track, since they are moving/changing state. In particular, our annotation guidelines requested annotators to list (a) objects that the camera-wearer interacts with through their body or tools; (b) other objects that are relevant to the activity (e.g., supporting surfaces like kitchen top); and (c) body parts (hands and legs). Note that every time an object changes visual state (adopting the Point-of-No-Return definition from [47]), it is marked as a new object (e.g., annotators list *tomato* and *sliced tomato* as two distinct object instances).

Which objects to annotate with masks? For scenarios that involve few objects (Music, Basketball and Soccer), we annotated all object instances. Instead, for Cooking, Health and Bike Repair we sampled object instances based on their frequency of occurrence and their size, due to time and budget constraints. In particular, we binned each object annotated in the Object Enumeration stage into bins based on their frequency of occurrence across the dataset (high, low) and object size (small, large). We then uniformly sampled object instances from these bins while accounting for annotation time and budget and proceeded with segmentation mask annotations. We ignored all objects with area < 150 pixels. For Cooking, specifically, we also filtered out a few objects such as spices, mixtures and liquids, as they tend to be too small to match in the exo view. Finally, we skipped exocentric mask annotations for objects that were visible in fewer than 10 frames of the egocentric video.

What frame rate to annotate at? We annotated segmentation masks at 1 frame per second, except for videos from the *Music scenario* which we annotated at 0.1 fps due to extremely long video durations.

In total, our annotation process yielded segmentation masks for 5,566 objects in 1,335 ego-exo video-pairs. Approximately 4M million frames were annotated resulting in a total of 742K ego and 1.1M exo paired segmentation masks. Apart from this we also annotated 367K ego only segmentation masks. Collectively this results in a total of 2.2M segmentation masks. Table 8 shows a detailed breakdown per scenario for the paired masks.

Formal task definition Given a pair of time-synchronized egocentric and exocentric videos, as well as a query object track in one of the views, the goal is to output the corresponding mask for the same object instance in the other view for all frames where the object is visible in both views. This task is especially challenging in our dataset, since we have to handle long videos with an average length of 3 minutes, as well as very small objects with areas of only a few pixels. Importantly, note that the input to the model *excludes* semantic labels or names for the objects, camera pose information relating the two views, and IMU or active range sensor measurements. We do not use such information as we want to encourage the development of methods for open-world correspondence, not relying on predefined sets of objects or inputs that require non-consumer camera devices.

Metrics We adopt the following metrics in our evaluation:

1. *Location Error* (LE), which we define as the normalized distance between the centroids of the predicted and ground-truth masks.
2. *Intersection Over Union* (IoU) between the predicted and ground-truth masks.
3. *Contour Accuracy* (CA) [117], which measures mask shape similarity after translation is applied to register the centroids of the predicted and ground-truth masks.
4. *Visibility Accuracy* (VA) [16], which evaluates the ability of the method to estimate the visibility of the object in the target view, as in practice it may often be occluded or outside the field of view. We measure this performance using balanced accuracy. Note that, in contrast to the previous metrics that compare segmentation masks at frames where the object is visible in both views, this metric is computed based on all frames with query masks.

Baselines Finding object mask correspondences across pairs of videos is an under-explored area in video understanding. Therefore, we investigate two diverse baseline approaches for our ego-exo correspondence task: (a) a *spatial model* that tackles the correspondence problem independently at each time point, and (b) a *spatio-temporal model* that takes into account the history of predicted correspondences.

Spatial baseline model. This model receives as inputs an egocentric frame, the associated exocentric frame, and a query object segmentation mask in one of the views. It then outputs the mask in the other view (if the object is visible in that view). It can be thought of as a generalization of query-point correspondence approaches proposed for sparse image correspondence [65]. We implement this baseline in the form of a Transformer-based image correspondence model,

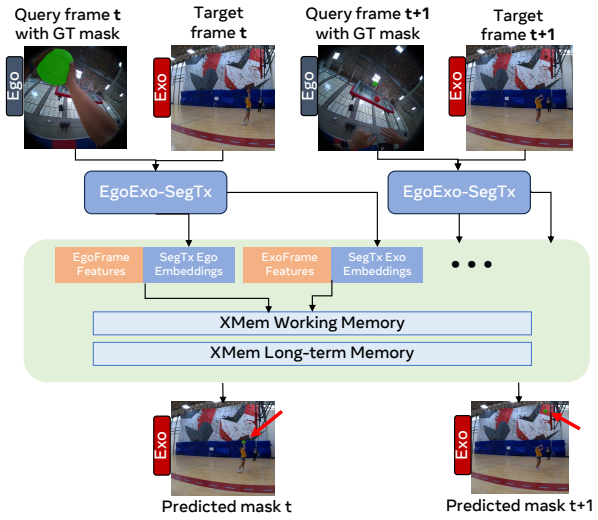


Figure 27. Overview of our spatio-temporal XView-XMem baseline model for the correspondence task.

XSegTx (Cross View Segmentation Transformer), which extends SegSwap [142], a method originally proposed for image co-segmentation, i.e., for segmenting common objects in a pair of images. To adapt the architecture of SegSwap for our correspondence problem, we additionally condition the model on the segmentation mask of the object of interest by feeding the query mask as a third input to the model. In particular, we first pass the egocentric frame, the exocentric frame and the query mask (as a binary mask) through a visual backbone network. We then flatten the resulting features into three sequences and pass them through the cross-image transformer with alternating self-attention and cross-attention layers. We first use the query mask features to attend to the features in the query view which are then used to cross-attend over features from the target view. This allows the model to reason over features from both views conditioned on the input mask. The resulting sequences for both views are "unflattened" and passed through a decoder to predict object segmentation masks in both views. We also pass the target view features through a classification head to classify if the query object is visible in the target view.

We train the model to perform mask prediction using a point-wise binary cross-entropy loss and a dice loss over the predicted and ground truth masks. We use only pairs of frames where the object of interest is visible on both views and apply the losses on predicted masks in both the views. During inference, we only consider the mask predicted in the target view and discard the predicted mask in the query view. We train the head performing Visibility classification using a binary cross-entropy loss on all the frames of the sequence.

Spatio-temporal baseline model. The spatio-temporal model receives as input the pair of ego-exo video clips as well as an object segmentation track in one of the views, and outputs segmentation masks in the other view for the frames that the object is visible in both views. It can be thought of as performing generalized tracking across views. We build our baseline model on top of XMem [24], a model originally proposed for tracking a specific target object given its segmentation mask in the first frame. In particular, our baseline model, called *XView-XMem*, adapts XMem to track the object across different views given ground-truth segmentation masks for one of the views in each frame. To encourage the model to learn associations of the objects between egocentric and exocentric views, we train XView-XMem to track the object in a sequence of interleaved frames of egocentric and exocentric views, i.e., each egocentric frame is followed by an exocentric frame and vice versa, as shown in Figure 27.

To mitigate track drift (within and across views), we also explore feeding the XSegTx embeddings to the XMem working memory. Since these embeddings are trained to guide the mask decoder at each frame independently, they capture rich information about the object of interest. The extracted image features from the ResNet in XMem are fused with the encoded embeddings from multiple layers of SA (self-attention) and CA (cross-attention) layers of XSegTx. They are then projected into keys and stored in memory for tracking.

Implementation details For our spatial baseline model, we downsample the images to 480x480 resolution for all the views while using padding to keep the original aspect ratio of the images. For the image backbone we use the same ResNet50 [51] checkpoint as SegSwap and freeze its weights during training. Our cross-image transformer architecture also follows [142]. We use a batch size of 32 and Adam [69] as our optimizer with a learning rate of 0.0002 which decays to 0.0001 after 50,000 iterations. We run all our experiments on a single Nvidia RTX A6000 GPU for 200,000 iterations.

For our spatio-temporal baseline model, we use the same visual backbone (ResNet50 [51]) and architecture as XMem [24]. Our only modification is in the information that gets inserted in the working memory at each frame. We first extract features from both ResNet and XSegTx for both both query and target frames. The corresponding features are then concatenated and projected to the original feature dimension through simple 2D convolution. We train on sequences of 8 interleaved ego and exo frames. The model is trained using AdamW as our optimizer with a learning rate of 0.00001 for 50,000 iterations and weight decay 0.05. The batch size is 8 clip pairs. We initialize our model with the original pretrained XMem, and keep both the ResNet backbone as well as our finetuned XSegTx models frozen. Note that we do not apply any data augmentations.

Data We use 1028 takes from the Ego-Exo dataset to train and evaluate models for this benchmark. In particular, we use the common split shared across benchmarks, with 657 takes for training, 156 takes for validation and 215 takes for testing. We extract pairs of images between egocentric and exocentric views which have corresponding object masks annotated for training. This gives us a total of about 193k pairs for training.

Results We benchmark our XSegTx and XView-XMem baseline models on the test set in Table 9. We experiment with two settings: providing the ground-truth object track in the exo view (exo query mask) and predicting it in the ego view, and vice versa.

First, we observe that exploiting temporal cues helps with tackling the object correspondence task as shown by the significant increase in performance achieved by the spatio-temporal baselines (ST type) compared to the spatial ones (for example, IoU improves from 14.6% to 21.1% in the Ego$\rightarrow$Exo setting.).

Second, we can see a big difference in performance between the Ego$\rightarrow$Exo and Exo$\rightarrow$Ego settings for all the baselines. In particular, models perform worse when the sequence of query masks is provided for the egocentric video and the model needs to predict query masks in exocentric video. This might be due to the heavy occlusion and very small size of objects in the exocentric views, making segmentation very challenging. While predicting a very tiny mask in the exo view can be very difficult, models can reason about the type and rough location of the object from a tiny mask in the exo view and thus accurately detect and segment it in the ego view, where it is much larger.

However, still all our baselines achieve a performance of $\leq$ 21% IoU in the Ego$\rightarrow$Exo setting and $\leq$ 59% IoU in the Exo$\rightarrow$Ego setting showing the challenging nature of the task and the dataset. We note that our dataset includes a great degree of object shape variation and high number of very small objects which are very difficult to model.

We break down our results across different activities in Fig. 28. We note that some activities are generally easier to model (e.g., basketball, soccer) because of limited variation in object shape and appearance whereas some activities (e.g., cooking and bike repair) are much harder to model due to larger diversity in appearance, shape and size of the objects across views. We also explicitly evaluate our baselines on their ability to predict masks for very small objects. To do so, we split our validation set based on the predicted object size in proportion to pixels in the image. We see that, all our baselines struggle on very small objects and perform increasingly well on larger object sizes.

Query Mask	Method	Type	Vis. Acc.↑	IoU↑	Location Error↓	Contour Acc.↑
Ego	XSegTx (random weights)	S	50.00	0.60	0.116	0.017
Ego	XSegTx	S	62.63	13.88	0.154	0.239
Ego	XMem (w/o finetuning)	ST	34.50	4.62	0.164	0.065
Ego	XView-Xmem (w/ finetuning)	ST	**92.70**	14.60	0.160	0.227
Ego	XView-Xmem (+ XSegTx)	ST	70.50	**21.10**	**0.100**	**0.323**
Exo	XSegTx (random weights)	S	50.00	1.62	0.197	0.027
Exo	XSegTx	S	74.60	21.80	0.133	0.265
Exo	XMem (w/o finetuning)	ST	78.30	43.80	0.103	0.446
Exo	XView-Xmem (w/ finetuning)	ST	**99.10**	43.40	0.112	0.448
Exo	XView-Xmem (+ XSegTx)	ST	95.80	**59.20**	**0.066**	**0.638**

Table 9. Baseline evaluation on the correspondence benchmark. Best results are reported in bold, second best results are underlined.

We also show some qualitative results in Fig. 30. As we can see, the spatial baseline (XSegTx) struggles to track the same object throughout the video. For example, in the bottom example, XSegTx alternates between predicting one and two object masks whereas the spatiotemporal baseline (XView-XMem) reliably tracks a single object throughout the sequence, showing the importance of exploiting temporal cues in the data.

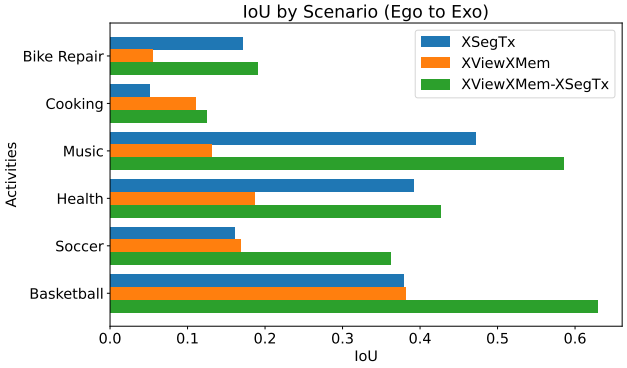


Figure 28. Performance of both baselines per activity scenario.

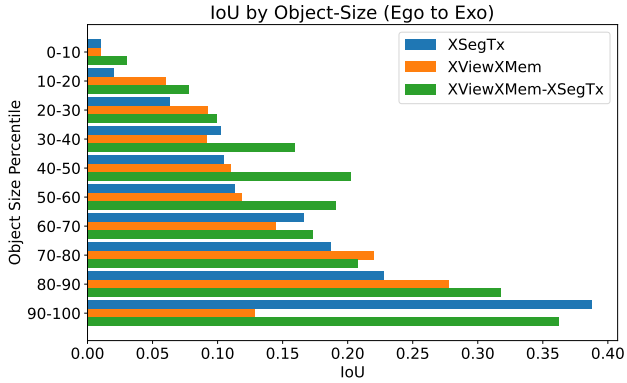


Figure 29. Correspondence evaluated across different object sizes in the target (exo) view. The object sizes range from $7e^{-6}\%$ to 11% pixels in the target view.

Contributions Statement Manolis Savva co-led the correspondence benchmark and contributed to the task definition, the annotation guidelines, the baseline design, and paper writing. Effrosyni Mavroudi co-led the correspondence benchmark and contributed to the task definition, the annotation guidelines, the baseline design, and paper writing. Lorenzo Torresani contributed to the task definition, and to editing this section. Sanjay Haresh developed the spatial baselines and contributed to data analysis, experimental results, and paper writing. Yongsen Mao developed the spatiotemporal baselines and contributed to data analysis, experimental results, and paper writing. Suyog Jain formulated the annotation pipeline, developed annotation tools and contributed to annotation guidelines and paper writing. Santhosh Ramakrishnan contributed to the annotation guidelines and the formulation of the annotation pipeline. Xitong Yang contributed to the annotation guidelines and the task definition. We would like to acknowledge Hanxiao Jiang for helpful discussions and preliminary ideas on baseline implementation. Devansh Kukreja built the render flow to generate frame-aligned videos of each camera for each take as model input.

13.A.2 Ego-Exo translation

Annotations Translation uses the same annotations as the correspondence task.

Formal task definition The translation benchmark focuses on generating information in the egocentric view given the exocentric view. The task is separated into two subtasks (see Fig 4): 1) *Ego Track Prediction* requires predicting the segmentation mask of an object in the unobserved ego frames given the object masks in an observed exo clip; 2) *Ego Clip Generation* entails generating RGB

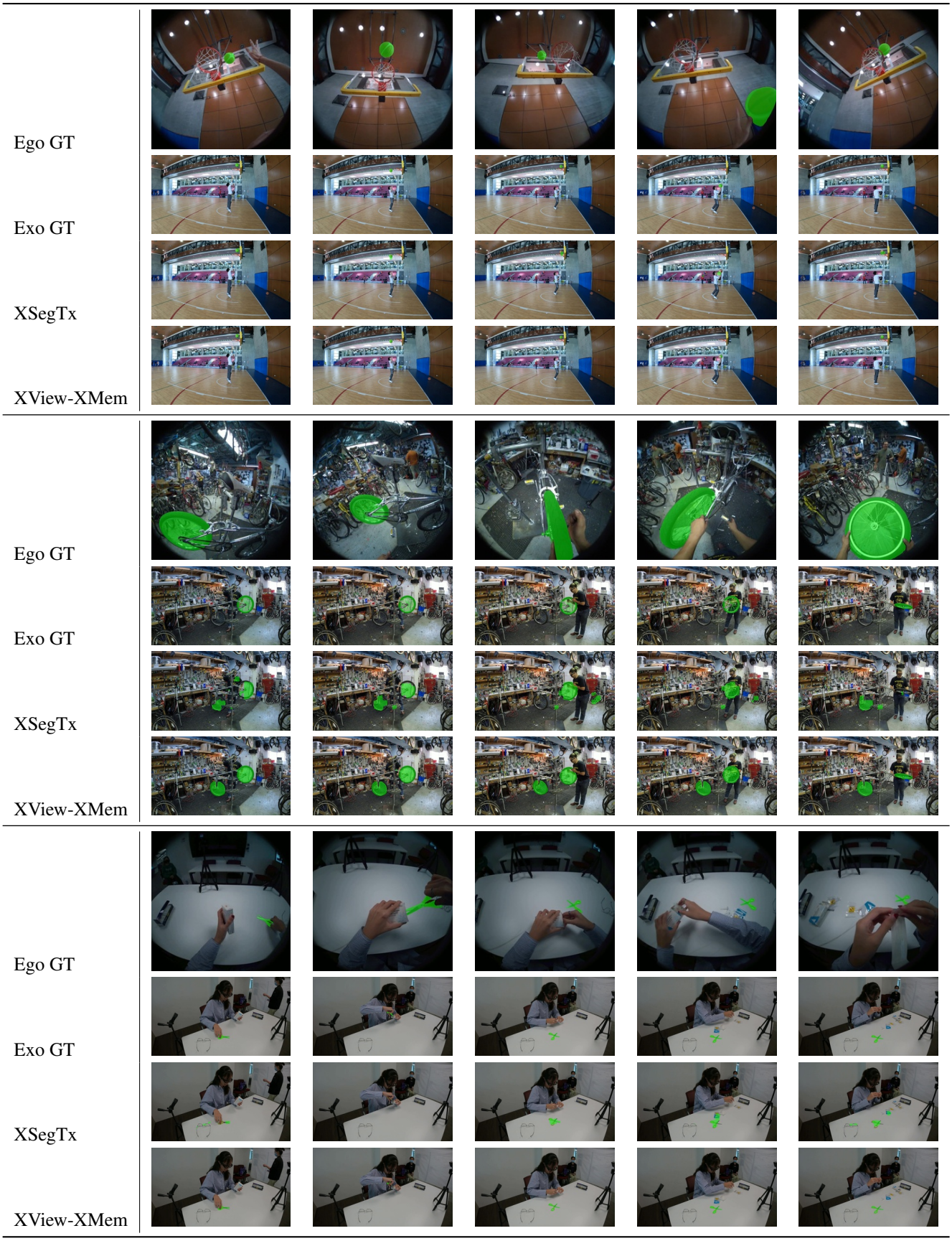


Figure 30. Qualitative results for the different correspondence baselines.

image values within the given masked ego view based on the exo view. For both subtasks, the input exo clip consists of 5 frames evenly sampled from a time span of 5 seconds.

Note that we restrict the input to include only the exo view and the object masks in order to promote the design of methods that can translate arbitrary third-person video into an egocentric one. Thus, the input *excludes* depth maps, 3D point clouds, IMU, or SLAM, which would simplify the task at the expense of general applicability, since these signals are typically not available for in-the-wild video. The only exception is a variant of the task where the ego camera pose for all frames of the clips is given as input. We consider this formulation in order to estimate a sort of "upper bound" on translation performance under the unrealistic assumption of known ego-exo camera relation.

Metrics We adopt a diverse set of metrics to assess the different aspects of the generated translation. As for the task of correspondence, we use *Visibility Accuracy* (VA) to evaluate the ability of the method to predict the visibility of the target object in the ego view but this time given only exo frames as input. Furthermore, we adopt the following metrics defined for correspondence to gauge the performance of *Ego Track Prediction*: 1) *Location Error* (LA) 2) *Intersection Over Union* (IoU) and 3) *Contour Accuracy* (CA) [117]. The IoU and CA are calculated after registering the centroids of the predicted mask and the ground-truth ego mask, in order to gauge mask prediction independent of location error. To evaluate *Ego Clip Generation* we use two popular image quality metrics (SSIM, PSNR [54]) and three perceptual metrics (DISTS [33], LPIPS [192] and CLIP similarity [124]).

Baselines For track prediction, we implement the GAN-based method pix2pix [61] and the NeRF-based method GNT [154]. For clip generation, we employ the GAN-based method pix2pix [61] and the diffusion model DiT [116]. It is worth noting that, as discussed below, we introduce specific modifications to adapt these methods to our task requirements. All baselines utilize exo images and masks, with only the GNT model making use of the extra input of ego camera pose.

Ego Track Prediction involves generating segmentation masks for the egocentric view based on the exocentric video clip and the exocentric object masks. We consider the following two baselines for this task:

- **pix2pix-mask**. We modify the generator of pix2pix to have inputs and outputs of 4 channels. Specifically, the exo frame and the exo mask are concatenated as the inputs while the 4-channel outputs are ego frame (3 channels) and ego mask (1 channel). The ego frame is supervised with the losses used in pix2pix. We use the bootstrapped cross-entropy loss [129] and the dice loss [152] for mask prediction.
- **GNT-mask**. We adopt the Generalizable NeRF Transformer (GNT) [154] as another baseline leveraging the camera poses. In our adapted version, the image encoder takes a 4-channel image (exo frame and mask) as inputs to predict the ego frame and ego mask. Formally, during the training of our GNT-mask, for each point x and viewing direction unit vector $d \in \mathbb{R}^3$, the ray transformer f in GNT predicts two key attributes: RGB Color (c) and Object Existence Score (e), in which e signifies the probability of an object being present at point x. During rendering, the volumetric radiance field encoded by the ray transformer can then be rendered into a 2D image as well as a 2D object mask.

Ego Clip Generation requires producing pixel values representing the target object in the egocentric view. To achieve this, we leverage 6 different input images for each frame: exo frame, exo mask, exo object crop, cropped exo mask, ego mask and cropped ego mask. The cropped exocentric and egocentric masks are generated by considering a bounding box to isolate the relevant portions of the exocentric and egocentric masks, respectively. The "exo object crop" refers to the RGB image obtained by cropping out the relevant region using the cropped exocentric mask. We resize these 6 images to the same size (256×256). We evaluate two baselines for this task:

- **DiT-pix**. We adopt the Transformer-based diffusion model DiT [116]. We predict the ego object crop by conditioning the DiT on the 6 input images in two manners. Initially, these six images are concatenated along the channel dimension and subsequently combined with the noisy ego object crop, forming the input to DiT. Additionally, two ResNet-50 architectures encode the six images into low-dimensional features, which are then incorporated into each layer of DiT via AdaLN [118].
- **pix2pix-pix**. We adopt pix2pix [61] for clip generation as well by concatenating the 6 images along the channel dimension as inputs to the pix2pix model.

All of the above-mentioned baselines perform image-to-image generation. We implement also clip-to-clip variants of these methods by taking multiple frames as inputs and predicting results for all frames jointly. For pix2pix, we achieve this by replacing the original 2D-Conv with 3D-Conv, and 2D-BatchNorm with 3D-BatchNorm. For DiT, we use space-time divided attention as in TimeSformer [13].

Results We employ the validation set for the purpose of selecting optimal checkpoints and hyper-parameters, which are subsequently evaluated on the test set.

In the context of Ego Track Prediction (Table 10), we

Method	Ego Cam. Pose	Location Error↓	Contour Acc.↑	IoU ↑
pix2pix-mask	No	20.7	4.1	4.6
+multi-frame	No	23.0	8.0	8.5
GNT-mask	Yes	**18.5**	**15.7**	**10.1**

Table 10. Evaluation of translation baselines for the subtask of ego track prediction.

Method	SSIM ↑	PSNR ↑	DISTS ↓	LPIPS ↓	CLIP ↑
pix2pix-pix	0.51	16.2	0.37	0.60	72.1
DiT-pix	0.56	16.0	0.31	0.47	83.6

Table 11. Evaluation of translation baselines for the subtask of ego clip generation.

gauge performance using *Visibility accuracy* (VA) to assess the ability of the model to estimate the object visibility in the ego view. Correctness is determined when two conditions are met: (1) the predicted mask is empty when the object is invisible in the ego view, and (2) the predicted mask is non-empty when the object is visible in the ego view. Both pix2pix-mask and GNT-mask fail to perform well in estimating the object visibility, achieving Visibility accuracy about the same as the 50% performance of random guess (55.6% for GNT-mask and 47.6% for pix2pix-mask). However, the ResNet-50 trained exclusively to attend to this binary classification achieves a VA of 80.6%. We assess mask quality by considering distance (Location Error) and similarity metrics (IoU and Contour Accuracy) between predicted and ground-truth masks after registration. The 3D-aware NeRF-based baseline, GNT-mask, outperforms the implicit baseline, pix2pix-mask, overall. However, it does so by exploiting the ego camera pose as additional input. It is noteworthy that both baselines perform poorly on this task, likely due to the inherent challenges in correctly predicting the location and shape of the target object in the ego view, probably due to the fact that it often has diminutive size in the exo view.

In the case of Ego Clip Generation (Table 11), the Diffusion model DiT-pix demonstrates superior performance across all metrics compared to the GAN-based pix2pix-pix. Qualitative results (Figure 31) illustrate that DiT-pix can generate highly photorealistic images, aligning closely with the ground-truth in most instances. However, there are occasional cases (the last 2 rows) where the shape of the object is accurately generated, but the texture deviates slightly.

We further verify the importance of each input in Figure 32. Without exo object crop as input, the model fails to correctly infer the color and texture of the target object in the ego view. This result is expected as the source objects often represent a very small region of the entire exo frame.

Scenario	IoU (%) ↑	LPIPS ↓
Basketball	16.2	0.38
Soccer	17.0	0.45
Music	4.3	0.28
Health	11.7	0.41
Bike	5.9	0.51
Cook	9.5	0.49

Table 12. Breakdown of results per scenario for the subtasks of ego track prediction (IoU) and ego clip generation (LPIPS).

Additionally, without the ego crop mask as input, the model predicts the orientation of the object incorrectly. These observations highlight the importance of the cropped inputs.

We can observe in Table 10 that multi-frame (i.e., clip-to-clip) prediction does not provide a quantitative advantage over frame-to-frame prediction. Yet, we noticed that the multi-frame variant often yields generations that are more consistent across frames, even for frames where the exo view is heavily occluded, as can be seen in Figure 33. This is reasonable as a clip-level model can more effectively learn about the target object from multiple frames and fill-in information that is missing in individual exo frames.

In Table 12 we provide a break down of the results across different scenarios, using GNT-mask for track prediction and DiT-pix for clip generation. We can observe similar trends for the two subtasks: the methods achieve better results in basketball and soccer scenarios than in bike and cook scenarios, which is reasonable as the objects in bike and cook scenarios are more complex and diverse.

Contributions statement Lorenzo Torresani co-led the translation benchmark, developed the task formulation, advised the baseline development, and contributed to editing this section. Judy Hoffman co-led the translation benchmark and advised the baseline development. Feng Cheng led the baseline development and implemented the pix2pix and DiT models for track prediction and clip generation. Mi Luo implemented the GNT baseline and the evaluation pipeline for ego track prediction. Ziwei Zhao contributed the pix2pix baseline for multi-frame input and led the evaluation for ego clip generation. Huiyu Wang advised the baseline development and contributed to the task definition, the baseline design, and the metric selection and analysis.

13.B. Ego-(Exo) Keystep Recognition

This family of tasks centers around recognizing the keysteps of a procedural activity and modeling their dependencies. Specifically, there are three tasks: fine-grained keystep recognition (Sec. 13.B.1), efficient multimodal keystep recognition (Sec. 13.B.2), and procedure understanding (Sec. 13.B.3). We refer to the family of tasks as "ego-(exo)" since exo may be available at the time of training but not inference.

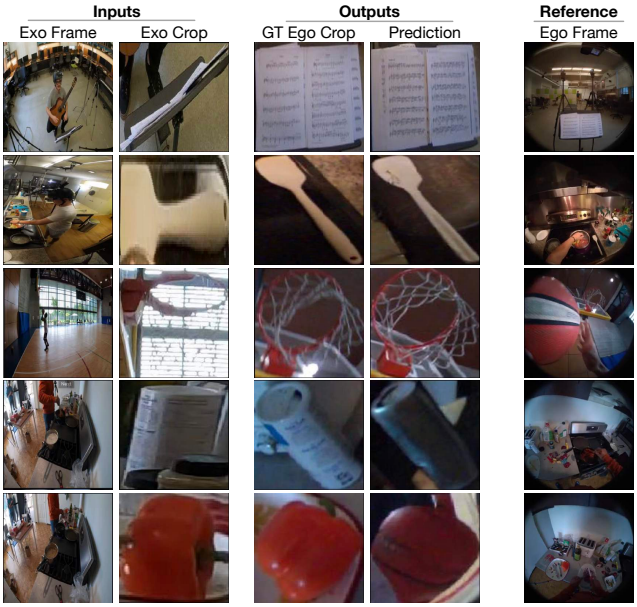


Figure 31. Qualitative ego clip generation by DiT-pix on the test set. The model takes 6 input images (exo frame, exo crop, exo mask, exo crop mask, ego mask, and ego crop mask). Note that only the exo frame and the exo crop images are included in this visualization. The ego frame, serving solely as a reference, does not constitute either an input or an output element.

Figure 32. Qualitative demonstration of the importance of the different inputs given to DiT-pix. The exo crop image and ego crop mask are critical for good performance.

13.B.1 Fine-grained keystep recognition

Annotations We annotate videos featuring any of the three procedural activities (i.e., cooking, bike repair, health) with temporal segments of *keysteps*, i.e., actions that contribute towards the completion of a procedural task. To accurately model the hierarchical nature of the activities, we also develop a data-driven hierarchical keystep taxonomy concurrently with the annotation process.

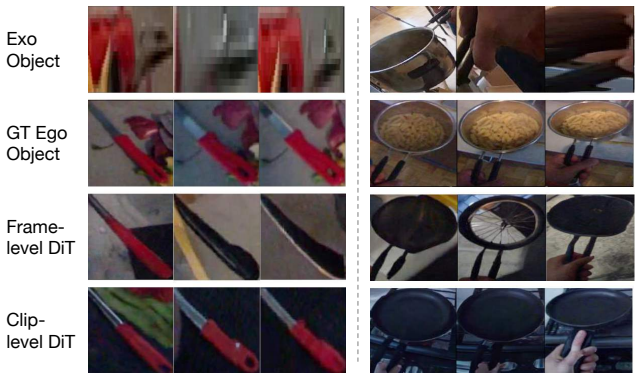


Figure 33. Comparison of ego clip generations using frame-to-frame vs clip-to-clip variants of DiT-pix. The clip-to-clip version of the model produces outputs that are more coherent across the frames of the clip, even for frames where the exo view is heavily occluded.

Figure 34 shows the annotation user interface. We provide annotators a composite view of time-synchronized ego and exo videos. Each keystep annotation contains the start and end timestamps, a category label, a natural language description, and a flag indicating whether the keystep is essential or optional for task completion. Annotators interact with a search widget which displays keystep labels with their complete path within a hierarchical tree, e.g., Making cucumber & tomato salad $>$ Prepare dressing $>$ To a bowl or jar $>$ Add salt.

As the activities performed by the camera wearers are unscripted, it is not possible to establish a comprehensive keystep taxonomy prior to annotation. To address this challenge, we designed an iterative, data-driven process for taxonomy development. We first initialize the taxonomy using various resources including recipes and instruction articles from the Web. This initial taxonomy captures keysteps that are generally expected in the activities, but it is assumed to be incomplete for the specific variations the camera wearers performed in the recordings. Subsequently, in each iteration, annotators receive the current taxonomy and are instructed to add new keysteps when they encounter actions not represented in it (see Figure 35). Any newly added keysteps are kept valid only for the duration of each annotation session and are not visible in other sessions. After a batch of videos have been annotated, we review the newly added keysteps to ensure their validity and update the taxonomy before repeating the process. We finalized the taxonomy after the third iteration, after which we re-annotated the entire set of videos with the final taxonomy for consistency.

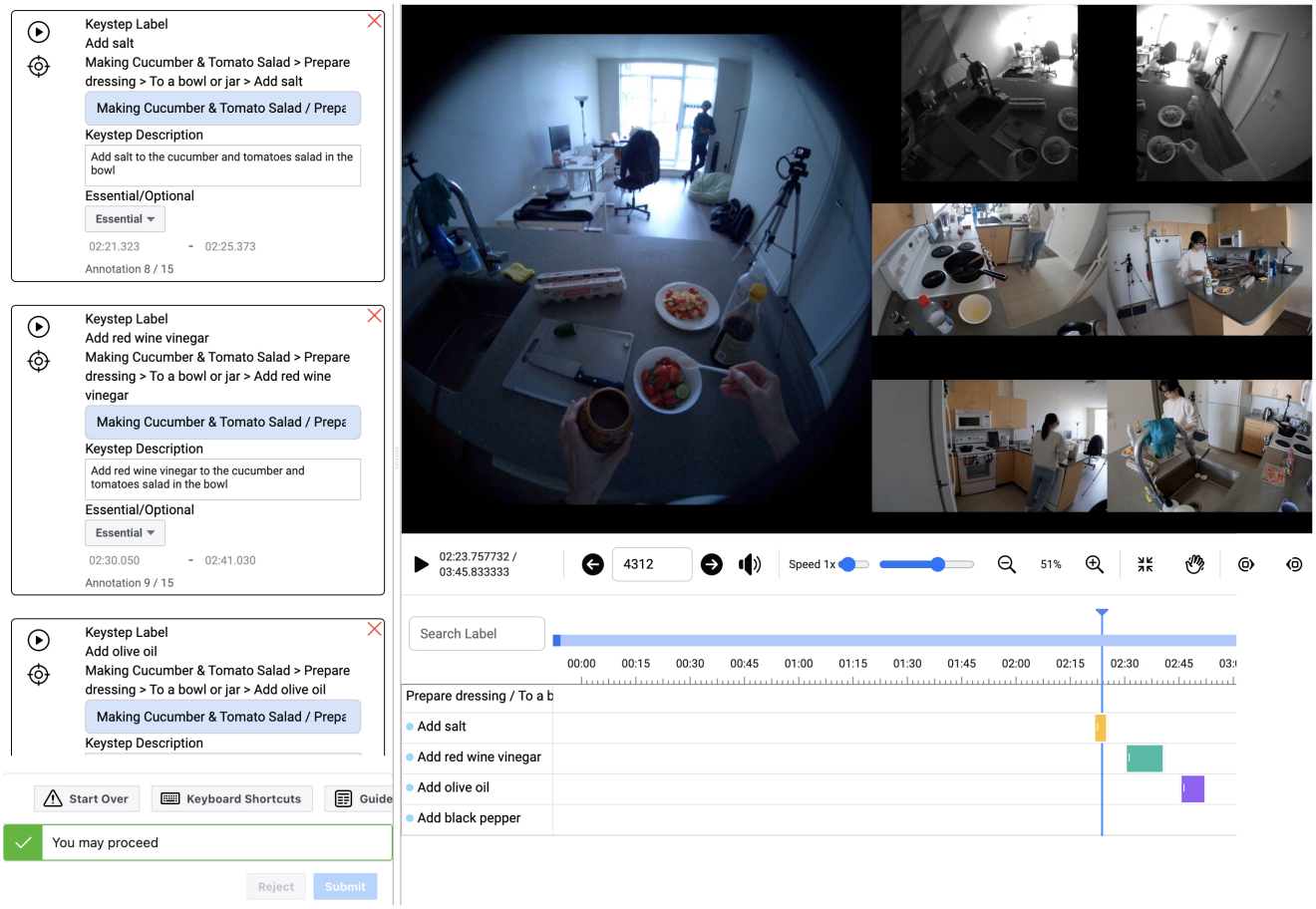


Figure 34. **The keystep annotation tool** shows a composite view of the time-synchronized ego-exo videos and the keystep time segment annotations. Each annotation consists of the start and end timestamps, a category label, a natural language description, and an essential/optional flag.

Scenario	Takes		Ego Keystep Segments		Ego + Exo Keystep Segments		Taxonomy	
	Count	Duration (total / avg†)	Count (total / avg†)	Duration (total / avg‡)	Count (total / avg†)	Duration (total / avg‡)	Activity	Keystep
Cooking	464	65.47h / 8.47m	19,034 / 41.02	58.08h / 10.99s	99,854 / 215.20	307.71h / 10.99s	11	527
Bike repair	293	13.51h / 2.77m	2,573 / 8.78	11.82h / 16.54s	12,865 / 43.91	59.12h / 16.54s	4	82
Health	331	18.72h / 3.39m	5,995 / 18.11	17.03h / 10.23s	30,723 / 92.82	86.99h / 10.23s	2	58
Total	1,088	97.71h / 5.39m	27,602 / 25.37	86.94h / 11.34s	143,442 / 131.84	453.82h / 11.34s	17	664

Table 13. **Keystep annotation statistics.** We report the statistics by grouping our 17 activities into three scenarios: cooking (11), bike repair (4), and health (2). Statistics are listed for takes† and keystep segments‡.

Formal task definition We consider trimmed video clip classification as the keystep recognition task. At training time we are given a labeled collection $\mathcal{D}$ of ego-exo video clips: $\mathcal{D} = \{(\mathcal{V}_{ego}^{(1)}, \mathcal{V}_{exo^{1-M}}^{(1)}, y^{(1)}), \ldots, (\mathcal{V}_{ego}^{(N)}, \mathcal{V}_{exo^{1-M}}^{(N)}, y^{(N)})\}$ where $y^{(n)}$ denotes the keystep label of the n-th sample. The video clips are manually trimmed from long procedural videos to contain only the keysteps to recognize. At test time, given *just* the ego view of a trimmed clip $\mathcal{V}_{ego}$, the model must predict its keystep label y.

Classification of trimmed video clips is a problem formulation commonly adopted in action recognition benchmarks [46, 67, 149]. However, our task differs from action recognition in two fundamental aspects. First, it targets fine-grained keystep recognition rather than classification of coarse activities. We note that this adds significant complexity, since different keysteps of an activity often involve manipulating the same objects in the scene (e.g., folding the bedsheet and smoothing out the bedsheet) and are consequently difficult to tell apart. Furthermore, different keysteps may be represented over largely different time spans (e.g., the average time span for "kneading dough" is

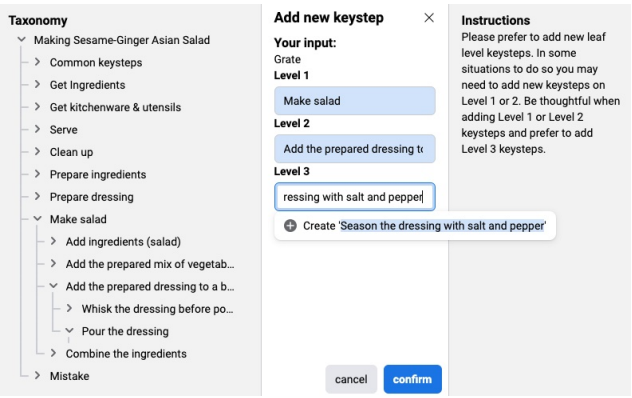


Figure 35. **Adding new keysteps to a taxonomy.** Annotators utilize a specialized widget to introduce new keysteps at any level within the existing taxonomy hierarchy.

87.3 seconds, in stark contrast with "getting salt", which averages at 3.6 seconds), thus requiring analysis at different levels of temporal granularity. The second key difference is the potential to leverage contextual cues available in exocentric videos during training to improve the prediction accuracy on egocentric videos. Note that at test time, the input to the model includes just the ego-view videos (RGB only). Exo-view videos, activity and scenario names, narrations, audio and associated metadata such as eye gaze, 3D point clouds, camera pose, and IMU information are *excluded* as inputs for inference (although we encourage exploring their potential utility in training) as our ultimate goal is a vision-centric approach that performs egocentric keystep recognition.

Baselines To understand the best strategy for egocentric keystep recognition with paired ego-exo training data, we consider a diverse set of baselines approaches, including methods for action classification, video representation learning, and ego-exo transfer.

- **Action classification.** As a prototypical example of this classic genre, we select a TimeSformer [13] model initialized with the checkpoint pretrained on the large-scale third-person action dataset Kinetics-600 [67] due to its strong performance in various video understanding tasks.
- **Video-language pretraining.** We adopt the EgoVLPv2 framework [123] and initialize the model with two backbones, one pretrained on the Ego4D dataset [47] (which contains only ego views) and the other that is further pretrained on EgoExo4D (which encompasses both ego and exo views), to exploit the variances in pretraining between these datasets.
- **View-invariant learning.** A two-stage training approach is employed. In the first stage, we utilize all available (ego, exo) video pairs in the dataset for training a view-invariant (VI) encoder. The training objective is a clip-level contrastive loss [110], aiming at identifying the synchronized (ego, exo) pairs as positive, and the non-synchronized pairs as negative. In the second stage, this pretrained model is further trained with a classification loss, aligning with the clip-level classification nature of the downstream task. Note that to align with the clip-level classification task, our contrastive loss operates at the clip-level, rather than at the frame-level as was done in view-invariant loss proposed in [140, 144].
- **Viewpoint distillation.** This also adopts a two-stage training approach. In the first stage, we train a multi-view teacher that takes both ego and exo views as input. In the second stage, a single-view ego student is trained, distilling knowledge [52] from the multi-view teacher to encapsulate information from both views.
- **Ego-Exo Transfer.** Here we follow the methodology proposed in Ego-Exo [82] which uses egocentric pseudo-labels to pre-train the network. We employ a masked autoencoder (MAE) [162] backbone, initialized from a Kinetics checkpoint, and the pseudo-labels provided from the Ego-Exo checkpoint to fine-tune with two auxiliary heads (Object-Score and Interaction-Map). We then further finetune the model with a classification loss for fine-grained keystep recognition.

For the first two baselines (which utilize pretrained checkpoints from well-established benchmarks), two training settings are further examined: one using only the ego view for classification loss and the other utilizing both ego and exo view videos, with the training objective being the sum of ego view and exo view classification losses.

Implementation details We use clips of size $8 \times 224 \times 224$, with frames sampled at a rate of $1/32$ for all baselines except for EgoVLPv2 (where we adhere to its pretraining scheme and sample 4 frames). The patch size is 16×16. For training, we resize the shorter side of the frame to a random value within the range of [256, 320], followed by randomly sampling a 224×224 region from the resized video. For evaluation, we sample a single temporal clip in the middle of the video, scale down the shorter spatial side of the video to 224 pixels and select 3 spatial crops (top-left, center, bottom-right) from the temporal clip to cover a larger spatial extent within the clip. The final prediction is derived by averaging the scores obtained for these 3 crops. We train our model for a total of 100 epochs on 4 NVIDIA V100 GPUs with a batch size of 32. The model checkpoint yielding the best performance on the validation set is selected and evaluated on the test set.

Data The keysteps in our dataset exhibit a very long-tailed distribution. To address this challenge, we set a cutoff threshold at 20 samples per keystep, limiting our analysis to

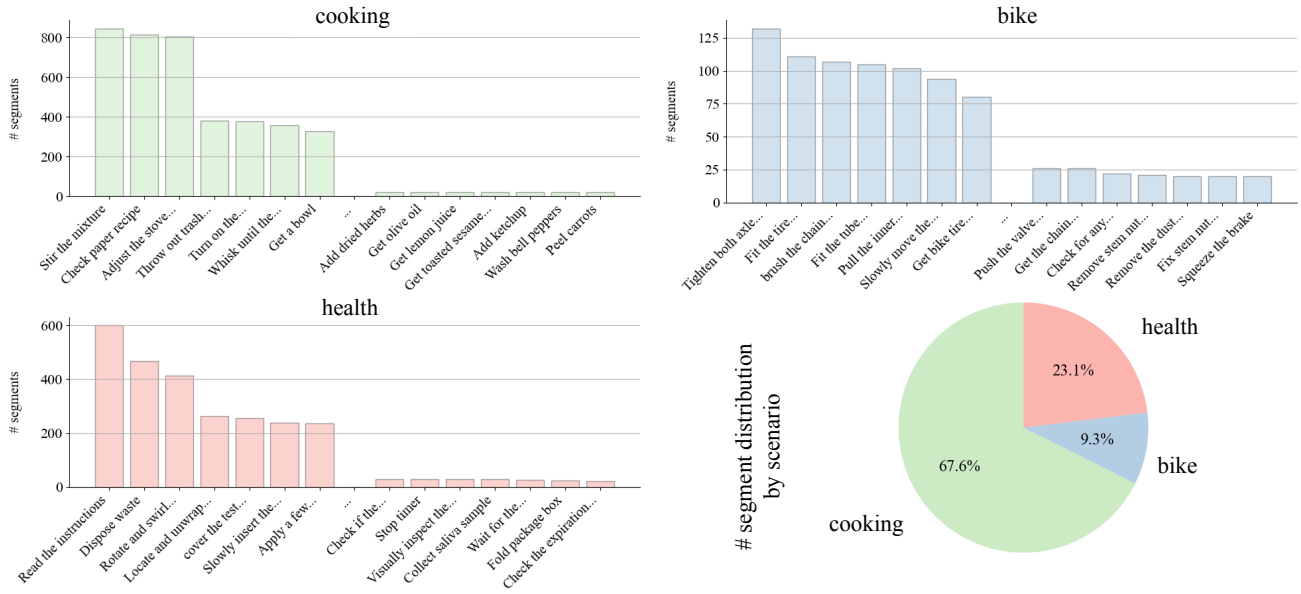


Figure 36. Keystep distribution in our dataset for each procedural scenario: cooking, bike repair, and health.

278 unique keysteps as shown in Figure 36. For simplicity, we consider only the leaf node keysteps in the hierarchy. Exploring the hierarchical structure including parent nodes is a promising direction but we leave this as future work. In all, the dataset for keystep recognition comprises 129,914 segments, with an average duration of 11.34 seconds each. Specifically, the training set contains 74,457 segments, of which 14,550 are from the ego view and the rest from the exo view. The validation set consists of 23,088 segments, including 4,502 ego-view segments, and the test set has 32,369 segments with 6,351 in the ego view.

		Ego Accuracy (%)	
Method	Train data	Val	Test
TimeSFormer [13] (K600)	ego	35.25	35.24
TimeSFormer [13] (K600)	ego,exo	32.67	29.84
EgoVLPv2 [123] (Ego4D)	ego	36.89	37.51
EgoVLPv2 [123] (Ego4D)	ego,exo	37.03	36.84
EgoVLPv2 [123] (EgoExo)	ego	37.61	37.85
EgoVLPv2 [123] (EgoExo)	ego,exo	38.21	38.69
VI Encoder [110] (EgoExo)	ego,exo	**40.23**	**40.61**
Viewpoint Distillation [52]	ego,exo	37.79	38.10
Ego-Exo Transfer MAE [82]	ego,exo	36.71	35.57

Table 14. Top-1 accuracy of keystep recognition. The pre-training dataset is denoted in parentheses. VI is short for view-invariant.

Results Table 14 reports the Top-1 accuracy for ego classification on both validation and test sets. Among all the baselines, the VI Encoder emerges as the top performer, achieving a test accuracy of 40.61%. It is closely followed by the EgoVLPv2 pretrained on EgoExo and Viewpoint Distillation, which attain test accuracies of 38.69% and 38.10% respectively. Additionally, we note that the MAE, when trained without Ego-Exo signals, recorded a test accuracy of 34.89%. These results open discussion on how to effectively utilize exo videos during training to enhance ego keystep recognition during test time.

First, we note that different approaches respond differently to the addition of exo-view videos during training. Specifically, while the TimeSFormer (K600) exhibits a degradation when the exo classification loss is integrated into the objective (i.e., test accuracy drops from 35.24% to 29.84%), EgoVLPv2 pretrained on EgoExo benefits from the introduction of exo-view videos (i.e., test accuracy improves from 37.85% to 38.69%). This enhancement is also evident in the VI encoder and viewpoint distillation when compared to TimeSFormer (K600) that only utilizes ego-view videos for training. These observations suggest that certain baselines are better equipped at leveraging exo information during training to improve ego keystep recognition.

We follow with a more detailed analysis of per-step performance in Figure 37, comparing training with ego-view videos and exo-view videos. We can observe that exo views show performance advantages over ego views in several steps, with the keystep 'have a conversation asking different questions' benefiting the most from exo. Conversely, ego views are more effective in steps involving manipulation of small objects, like 'cut carrots' and 'unpack the new

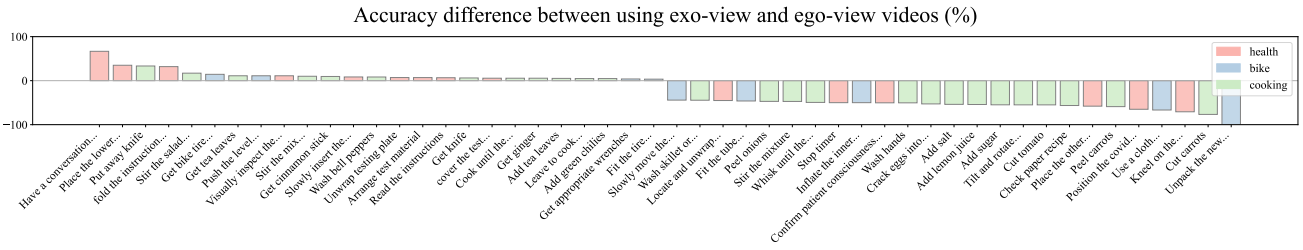


Figure 37. Keystep recognition evaluated per keystep label, comparing training with only ego-view videos versus exo-view videos. The accuracy delta (exo-ego) is displayed, where a positive value indicates better exo and a negative value indicates better ego performance.

tube'. This observation can be linked to the positioning of exo cameras, which are often placed further away from the subject, enabling them to capture a broader view, though possibly missing finer details of the activity. We hope these findings provide insight for future research on the effective use of exo-view videos during training.

Overall, we posit that the endeavor to enhance view-invariant learning and to more effectively harness the complementary information from exo views for ego keystep recognition remains an open avenue. Our findings underscore the need for further investigation and innovation in this domain.

Contribution statement Tushar Nagarajan co-led the keystep recognition benchmark, co-developed the task formulation, and advised the baseline development. Yale Song co-led the keystep recognition benchmark and led the keystep annotation effort, including design of annotation guidelines, taxonomy development and coordination of annotation workflows; he also contributed to the task formulation, advised baseline design, and facilitated the delivery of EgoVLPv2 pretrained backbone. Triantafyllos Afouras contributed to the taxonomy definition, managed the labeling effort, and developed software for post-processing the annotations. Zihui Xue led the baseline development effort, implemented the TimeSFormer, EgoVLP, VI Encoder and Viewpoint distillation baselines, and performed analysis of results. Eugene Byrne contributed to the taxonomy development and to the Ego-Exo Transfer baseline implementation and analysis. Avijit Dasgupta contributed to the annotation and taxonomy development, and to the early stage of baseline design. Miguel Martin contributed to the annotation and the taxonomy development. Shraman Pramanick contributed the EgoVLPv2 pretrained backbone. Yifei Huang contributed to the early stages of task definition and baseline design. Devansh Kukreja built the render flow to generate frame-aligned videos of each camera for each take as model input, and to produce video collages for annotations. Lorenzo Torresani contributed to editing this section. Kristen Grauman contributed to the task formulation.

13.B.2 Energy-efficient multimodal keystep recognition

Annotations and data This task uses the same egocentric videos and annotations as keystep recognition. However, in addition to the raw RGB video, it uses the audio stream (and potentially other sensors) as another sensor modality.

Formal task definition In this task, the goal is to perform *online* classification of keysteps in a streaming egocentric multi-modal video, within an energy budget. We consider an ego video $\mathcal{V}_{ego}$ of arbitrary length T comprising a stream of K different sensory modalities (*e.g.*, RGB images, audio, IMU, *etc.*). At each time step t, where $1 \leq t \leq T$, the video consists of samples for each available modality, such that $\mathcal{V}^t_{ego} = \{S^t_1, \ldots, S^t_K\}$, where S^t_j denotes the sample at time t for the j^{th} modality. Given $\mathcal{V}_{ego}$ and an energy budget B, our task is to learn a model $\mathcal{F}$ that maximizes the overall keystep recognition performance across the full video while also ensuring that the combined energy for sensing and running model inference does not exceed B. F consists of a sensor triggering policy $\mathcal{F}^{\mathcal{P}}$ and a keystep prediction model $\mathcal{F}^{\mathcal{K}}$. At every step t, the policy $\mathcal{F}^{\mathcal{P}}$ decides which sensor(s) to activate and sample from, in order to produce the model's current observation O^t, such that $O^t \subseteq \{S^t_1, \ldots, S^t_K\}$. Given O^t, the keystep predictor $\mathcal{F}^K$ outputs its estimate of the ground truth keystep for the current step.

The energy budget accounts for the cost of operations in each model forward pass, the cost of moving intermediate activations in and out of memory and the cost of the continuous operation of sensors, each having different cost profiles (e.g., IMU and audio sensors are relatively cheaper to operate than camera sensors). Note that the sensor triggering policy may be static (e.g., sample video at 4 frames per second (fps), keep audio/IMU off; sample 1 fps video, keep IMU always on) or dynamic (e.g., depending on the audio, decide whether to trigger video capture). We keep our task definition general, allowing the challenge to admit a wide variety of recent approaches ranging from pure video-based efficient backbone architectures [41] to multi-

modal triggering approaches and, naturally, a combination of them.

Note that at test time, the input to the model can only include current and past observations as our task is strictly an online recognition task. However, we encourage exploring other modalities than those considered in our experiments, e.g., IMU or camera poses inferred from video, audio, and IMU.

Measuring energy consumption Accurately measuring energy consumption of models is crucial for their use in AR/VR devices [2, 23]. The energy used comes from a complex interplay of sources including sensors, compute, communication, data processing, memory transfer (SRAM and DRAM), and leakage – many of which are typically ignored when building *efficient* computer vision models, despite their large energy consumption (e.g., memory transfer accounts for over 50% of the total power [183]).

We consider three key factors when modeling energy consumption following prior work [153]. (1) Compute energy: the cost of each model forward pass as a function of the number of operations (MACs). (2) Memory transfer energy: the cost associated with memory read-write operations for storing intermediate activations and model outputs. (3) Sensor triggering energy: the cost associated with turning on / off and continuous operation of sensors (camera, audio, IMU). For a model that processes an observation O^t, the total energy consumed can then be formulated as.

$$E(O^t) = \alpha * C(O^t) + \beta * M(O^t) + \sum_{j=1\ldots K} \gamma_j * \mathbb{1}(S_j \in O^t), \quad (1)$$

where $C(O^t)$ corresponds to the total number of multiply-add operations computed during the forward pass (in MAC/s), $M(O^t)$ corresponds to the total memory transferred to/from DRAM (in MB/s), and $S_j \in O^t$ corresponds to whether the j-th sensor is active. Finally, α, β, γ_j are weighting factors that measure the contribution of each energy source. We select these weighting parameters to reflect real-world AR/VR hardware capabilities. Namely, α = 4.6 pJ/MAC [31, 153]; β = 80 pJ/byte [55]; γ_{rgb} = 15 mW and γ_{audio} = 0.5 mW [91].

While physical measurements are required for truly accurate energy estimates, this formula provides a reasonable approximation to it and can serve as a target for the computer vision community towards energy-efficient models for real-world devices.

Energy profiler We adapt off-the-shelf profiler software built for PyTorch to compute the total multiply-accumulate operations (MACs) and memory transfer (MB) required to estimate total energy in Eqn. 1. The quantities are time-normalized — total energy consumption is expressed as power (mW). We describe each component of the profiler below.

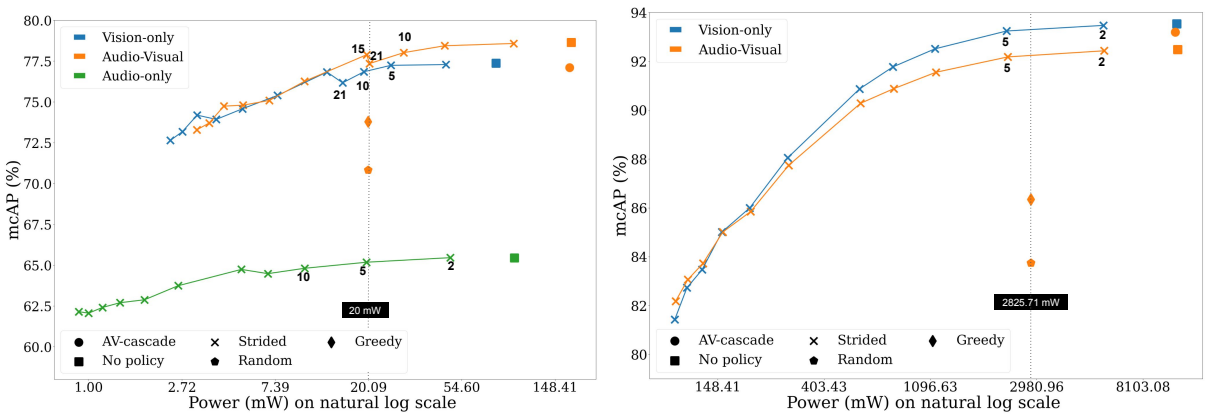
- **Compute operations (MACs)** We use the native PyTorch FLOP counter to get the total FLOP count in the forward pass. We convert this to MACs (approximately 2 FLOPs = 1 MAC).
- **Memory transfer (bytes)** We consider GPUs as our processing device, and use the PyTorch memory profiler to get the list of all operations executed in the forward pass (`model.forward()` call) and their associated GPU memory usage. The total memory is the sum of the individual operation memory costs.
- **Sensor capture** For each modality, we measure the time for which it is active as the number of observations sampled containing the modality. We require that the sensors capture at least 1 second worth of samples (roughly 100 samples) as energy consumption is ill-defined for an *instantaneous capture*.

Metrics Following prior work [30], we evaluate online keystep detection performance using per-frame calibrated mean average precision (mcAP), which accounts for the imbalance in the keystep labels in our dataset. We measure energy consumption in mW as described above.

There is a natural trade-off between efficiency and better performance. Thus, we evaluate models in two tiers by setting a budget for the power consumption in each tier, namely 20 mW for the *high-efficiency* tier and 2.8W for the *high-performance* tier. We select the *high-efficiency* budget based on the energy consumption of current single-modality, efficient architectures (e.g., X3D-XS [41]) with an eye to the future where multi-modal models operate within it. For the *high-performance* tier, we set the budget to a value that permits the use of powerful transformer-based action recognition models like LaViLa [199]. Once a model runs out of budget, in our setup it uses its latest prediction for all future steps.

Experimental setup We instantiate the task by considering keystep prediction episodes where the multimodal samples arrive in a streaming fashion. As mentioned above, we use vision and audio as our task modalities, where vision comprises RGB frames that are streaming at 30 frames per second (fps), and the audio modality is made up of time-aligned single-channel chunks that are 0.4 seconds long and sampled at 16 kHz. However, the setup can be extended to include IMU, and potentially other sensors as well. We evaluate all models at the rate of 5 fps on a total of 211 test episodes of variable length, where the shortest episode is $\sim$15 seconds, and the longest episode is $\sim$34 minutes.

Baselines We provide a family of (less/more expensive) keystep prediction models for solving the task. Each model



(a) *High-efficiency* tier (budget = 20 mW)

(b) *High-performance* tier (budget = 2825.71 mW)

Figure 38. Keystep prediction performance (mcAP) vs. total power consumption with different prediction backbones and sampling policies for both *high-efficiency* (left) and *high-performance* (right) tiers. For the models using a fixed stride, we show their stride value in text if their total energy consumption is close to the budget.

has a unimodal or audio-visual feature encoder followed by a keystep classification head.

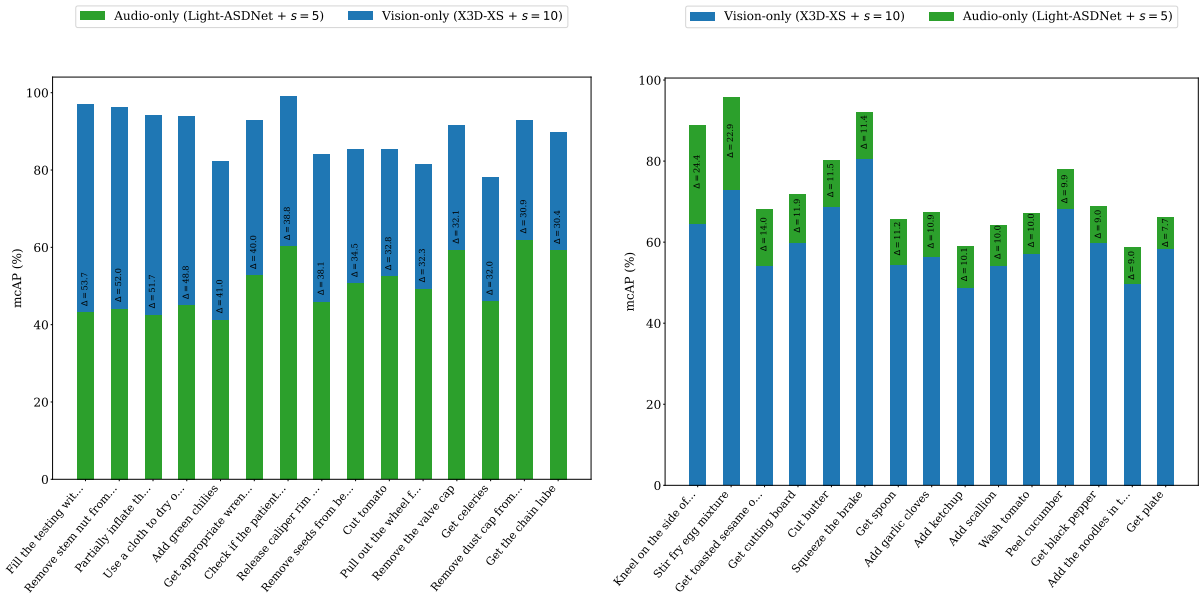
- **X3D-XS [41].** This is a vision-only model comprising the X3D-XS feature encoder, which progressively expands the feature size and representational capacity of its layers, and later contracts them for achieving better performance-efficiency trade-off. This is the most lightweight model in our family of keystep predictors. The encoder has a depth factor of 2.2, and takes 4 RGB frames of size 160×160 sampled at 15 fps, as inputs.
- **LaViLa [199].** This is another vision-only model where the visual feature encoder is trained through CLIP-style video-language pre-training. To improve the feature quality over vanilla CLIP-style pre-training, this method augments the number of video-text pairs by leveraging pre-trained large language models (LLM) to generate textual descriptions of un-annotated videos and rephrase existing narrations. In particular, we use the frozen TimeSformer [13]-Base (TSF-B) visual encoder pre-trained on the Ego4D dataset. To generate the feature for a target frame, the encoder samples 12 RGB frames of size 224×224 at 30 fps from a time window centered around the target frame and pads the samples with the boundary frames on both ends to create a 16-frame clip.
- **Light-ASDNet [83].** This is an audio-only model that represents audio as spectrograms and efficiently encodes them by splitting 2D convolutions into 1D convolutions along the spectrogram temporal dimension [83]. In our setup, the spectrograms are Kaldi [122]-compliant, and consist of 196 temporal windows and 160 Mel-frequency bins, respectively.
- **Audio-Visual Late Fusion (AV-LF).** This is an audio-visual model that does late fusion of visual features (encoded with X3D-XS or LaViLa) and audio features from Light-ASDNet by using linear layers.

To improve the energy efficiency of the aforementioned keystep predictors, we employ the following baseline policies for determining when to sample or skip each modality:

- **Fixed stride.** This is a policy that samples the input (video or audio) every s prediction steps. We evaluate different s values, where $s \in \{2, 5, 10, 15, 21, 43, 64, 86, 107, 129, 150\}$.
- **AV-LF + greedy.** This is a policy that greedily uses up the budget by sampling both audio and vision as early as possible, and uses the AV-LF backbone for keystep prediction.
- **AV-LF + random.** This is a policy that randomly samples or skips the audio and/or visual inputs until it runs out of budget, and uses the AV-LF backbone for prediction.
- **Audio-Visual (AV) Cascade.** This is a policy that initially uses the Light-ASDNet model to predict the keystep, and switches over to the LaViLa model (expecting a more accurate prediction from it) if the audio-based prediction confidence is below a certain threshold. We set the confidence threshold to 0.5 in our experiments.

Implementation details We train all keystep prediction models for 150 epochs using the cross entropy loss. We use the AdamW [95] optimizer with an initial learning rate of 10^{-4} and a weight decay of 10^{-5}. We set the batch size to 512 for vision-only models, and 384 for audio-only and audio-visual models.

Results In Fig. 38a, we plot the recognition mcAP of all models against their total power consumption for the *high-efficiency* tier. We can see that combining vision and audio is better than using only vision or audio. Thus suggests



(a) Vision improves over audio

(b) Audio improves over vision

Figure 39. Improvement (left) or degradation (right) in keystep recognition performance per keystep label, when comparing the most efficient vision-only (X3D-XS [41] + $s = 10$) and audio-only (Light-ASDNet [83] + $s = 5$) models from the *high-efficiency* tier. The plots show the 15 keysteps where improvement or degradation are largest. Δ reports the amount of improvement/degradation.

Method	Modality	mcAP (%) ↑	Power (mW) ↓
Light-ASDNet [83] + $s = 5$	A	65.18	19.67
X3D-XS [41] + $s = 10$	V	76.85	**19.14**
AV-LF w/ X3D-XS + $s = 15$	AV	**77.89**	19.70

Table 15. Keystep prediction results for the *high-efficiency tier* (budget = 20 mW).

Method	Modality	mcAP (%) ↑	Power (mW) ↓
Lavila [199] + $s = 5$	V	**93.24**	**2245.66**
AV-LF w/ Lavila + $s = 5$	AV	92.18	2274.40

Table 16. Keystep prediction results for the *high-performance* tier (budget = 2.8W).

that the two modalities carry complementary cues that are useful for the task. However, all vision-only models outperform their audio-only counterparts, which indicates that vision is the most critical modality for the task. The raw backbones generally perform better than the models using a sampling policy, but at the cost of requiring higher energy, making them impractical to use in online settings. Among the models that use a fixed stride, a lower stride generally improves the performance while hurting energy efficiency. For the vision-only and audio-only backbones, the best stride values within budget are $s = 10$ and $s = 5$, respectively. Among the audio-visual models, AV-LF with a stride of $s = 15$ performs the best among all models that satisfy the budget. Using the greedy or random policy with AV-LF leads to a sharp decline in performance compared to using a fixed stride, showing that sampling very early or randomly in the episode is suboptimal for our online recognition task. AV-cascade also performs worse than most audio-visual models while also requiring more energy, possibly because the audio backbone often outputs wrong but over-confident predictions that prevent switching over to the more reliable vision backbone when required.

For easy reference, in Table 15 we report the recognition performance and total power consumption of our best uni-modal and audio-visual models within budget for the *high-efficiency* tier.

In Fig. 38b, we plot the recognition mcAP of all models against their total power consumption for the *high-performance* tier. Different from the high-efficiency tier, the audio-visual backbone generally performs worse than the vision-only backbone, possibly because the LaViLa features are strong enough by themselves, and fusing them with audio features through the simple mechanism of linear late fusion reduces their expressivity. Otherwise, the overall behavior of different sampling policies is similar across the two tiers. We report the recognition performance and total power consumption of the best uni-modal and audio-visual models within the *high-performance* budget in table 16.

Additionally, in Fig. 39, we present a detailed analysis of the keystep labels where the best vision-only model from the *high-efficiency* tier yields the maximum improvement or decline in performance compared to its audio-only coun-

terpart. We observe that the vision-only model produces a large improvement over the audio-only model usually in steps where the activity does not produce distinctive sounds (*e.g.*, *add green chillies*, *get celeries*, *etc.*). On the other hand, using audio alone helps the most when the activities involve sounds that are strongly indicative of the nature of the task (*e.g.*, *stir fry egg mixture*, *cut butter*, *etc.*).

Finally, we envision that future work on this task will explore more sophisticated *learned* policies, potentially trained using reinforcement learning, in order to adaptively decide when to sample which modality instead of using fixed heuristics. Another promising direction is to investigate efficient transformer-based recognition backbones [180, 198] that can improve recognition performance without significantly affecting the model efficiency.

Contribution statement Tushar Nagarajan led the energy-efficient multimodal benchmark, co-developed the task formulation, and advised the baseline development. Sagnik Majumder led the baseline development effort and contributed all baseline implementations and analysis of results for the benchmark. Merey Ramazanova developed the energy profiler used to evaluate all baselines and contributed to the experimental analysis. Lorenzo Torresani contributed to editing this section. Mitesh Kumar Singh helped design the energy formula. Miao Liu and Shengxin Cindy Zha initiated this benchmark and developed an early version of the task formulation.

13.B.3 Procedure understanding

The real-world motivation for our procedure understanding task has basis in both augmented reality (AR) and robotics. AR assistants, beyond recognizing the current keystep, could verify missing mandatory keysteps, suggest possible future ones, and detect procedural mistakes. Similarly, robots could learn the structure of a procedure from human demonstrations. Mining the structure of procedures has been shown useful for planning [14, 21] and improving keystep recognition [9, 203] and discovery [10].

Annotations For each of the considered procedural tasks, we manually labeled task-graphs as structures encoding the keystep orderings leading to a correct execution of the procedure.

Task-graphs. We define a task-graph as a directed graph in which nodes represent keysteps and directed edges represent dependencies. For instance, in the task-graph reported in Figure 40, the "Add Milk $\rightarrow$ Get a Bowl" structure denotes that keystep "Get a Bowl" has to be executed before keystep "Add Milk". Besides directed

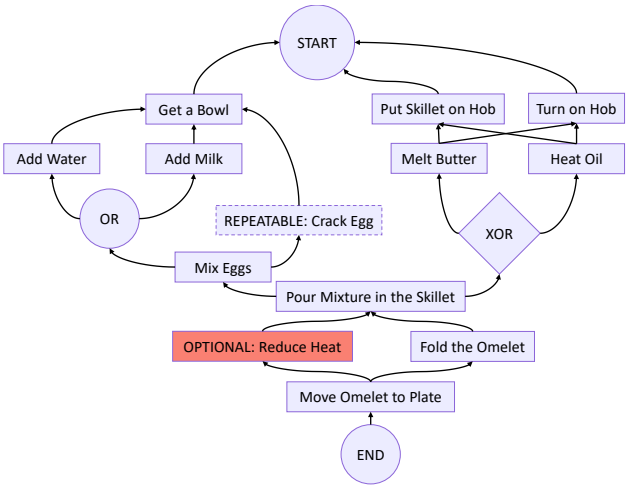


Figure 40. Example task-graph of a "Cooking Omelet" procedure.

edges, task-graphs also contain "OR" and "XOR" structures, which combine dependencies logically, as well as "optional" and "repeatable" node attributes.

Task-graph construction. We first familiarized ourselves with the procedural tasks by watching videos with annotated keysteps. We then initialized task graphs with procedural dependencies obtained from keystep annotations through the following procedure: a) a directed graph is first generated from the observed keystep transition frequencies; b) edges of the transition graph are filtered based on transition probabilities using a threshold parameter which is manually tuned for each scenario; c) edge directions are inverted to convert frequent transitions into dependencies. These initial graphs were then refined and corrected by human annotators leveraging common sense and in-domain knowledge.

Segment-level annotations. Let $S = \{s_1, \ldots, s_n\}$ be a labeled sequence of keysteps in a given video. We denote with y_i the annotated keystep label of segment s_i and with $Y_{:i} = \{y_1, \ldots, y_i\}$ the sequence of labels up to the i-th keystep. Using these keystep annotations, each segment s_i is automatically matched to a task-graph and augmented with the following attributes: 1) a list of *previous keysteps*—these are the in-neighbors of the matched node, 2) *optional* labels—directly derived from the optional node attribute, 3) a *procedural mistake* label—this is set to "true" if the in-neighbors of the matched node do not correspond to segments in the history $Y_{:i}$, 4) the list of *missing keysteps*—the in-neighbors of the matched node not listed in $Y_{:i}$, and 5) the list of *next steps*—nodes for which in-neighbors appear in $Y_{:i}$. Non-repeatable nodes are listed only if they do not appear in $Y_{:i}$.

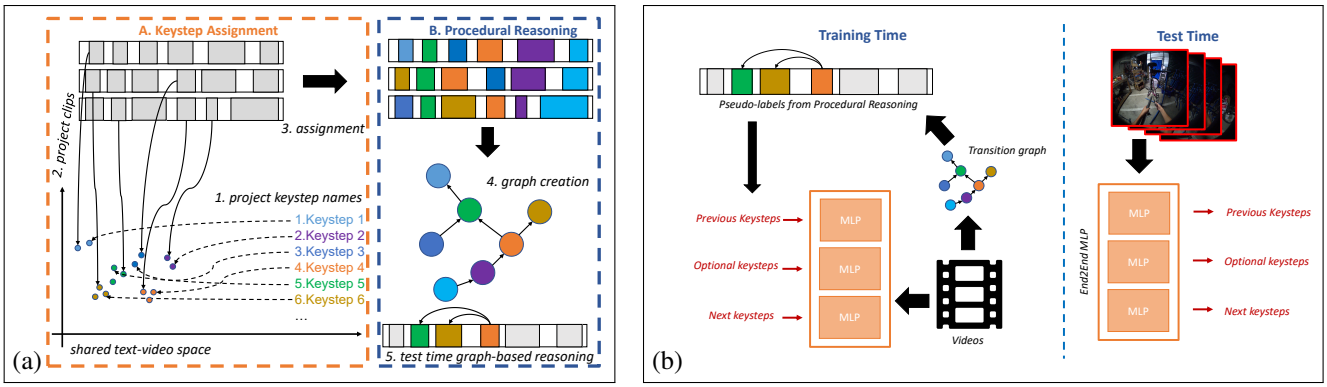


Figure 41. Overview of the two procedure understanding approaches considered in our evaluation: (a) graph-based baselines for procedure understanding rely on a Keystep Assignment and a Procedural Reasoning component; (b) the architecture of our end-to-end baseline.

Formal task definition Given a video segment s_i and its segment history $S_{:i-1} = \{s_1, \dots, s_{i-1}\}$, models have to 1) determine *previous keysteps* (to be performed before s_i); infer if s_i is 2) *optional* or 3) a *procedural mistake*; 4) predict *missing keysteps* (which should have been performed before s_i); and 5) forecast *next keysteps* (for which dependencies are satisfied and hence can be executed next). The task is weakly supervised, with two versions based on the level of supervision: 1) instance-level: segments and their keystep labels are available during training and inference; 2) procedure-level: unlabeled segments and procedure-specific keystep names are given for training and inference.

Note that, when the procedure-level supervision is considered, the input to the model *excludes* keystep labels both at training and test time. At both the procedure and instance levels of supervision, models are required to process the video in a causal fashion, meaning that predictions made at time t only depend on observations made at time $t' < t$.

Baselines We consider two main baseline approaches: graph-based baselines, which leverage an explicit procedure structure mined from videos, and an end-to-end model aimed to make predictions directly from the observed videos.

Graph-based baselines. Graph-based baselines are composed of a keystep assignment and a procedural reasoning component (see Figure 41(a)).

Keystep assignment is applied to obtain a pseudo-labeling of the provided video segments when the supervision is at the procedure-level (i.e., segments are unlabeled and only keystep names are provided). This is achieved by means of an EgoVLPv2 model [85] pre-trained on ego-exo videos and narrations. Video segments and keystep names are projected to the shared video-language space using EgoVLPv2. We hence assigned each video segment to the closest keystep in the representation space according to the cosine distance. In the problem formulation with instance-level supervision (i.e., when keystep labels are available for all segments during both training and testing), we use ground truth labels instead of those obtained from keystep assignment.

The procedural reasoning component creates for each procedure a transition graph based on keystep co-occurrences. In the graph, each node represents a keystep category, while directed edges represent the probability of transitioning from one node to another one. An edge $A \rightarrow B$ is assigned the following weight based on statistics collected from the training videos:

$$P(B|A) = \frac{\text{\# times keystep } B \text{ follows keystep } A}{\text{\# occurrences of keystep } A}$$

At test time, the graph is used to perform procedure understanding and answer the keystep-level questions. Specifically, given current segment s_i: 1) keystep y_{prev} is predicted as the previous keystep with confidence score equal to the transition probability $P(y_i|y_{prev})$, where y_i is the inferred or ground truth keystep label for segment s_i; 2) segment s_i is predicted as optional based on the empirical probability $\frac{\text{\# training videos containing } y_i}{\text{\# training videos}}$; 3) segment s_i is predicted as a procedural mistake with a score equal to the sum of the transition probabilities to y_i from keysteps y^{prev} that are missing from the keystep history, i.e., $\sum_{y^{prev}} [y^{prev} \notin S_{i-1}] \cdot P(y_i|y_{prev})$, where $[\cdot]$ is the indicator function; 4) keystep y is predicted as a possible missing keystep with probability $[y_i \notin Y_{:i-1}] \cdot P(y|y_i)$; 5) keystep y is predicted as a future keystep with probability $P(y_i|y)$.

End-to-end baseline. This baseline aims to provide an end-to-end approach to perform procedure understanding directly from the input clip. The baseline predicts previous keysteps, optional keysteps, and next keysteps by feeding

Supervision	Baseline	Keystep Labels	Prev. Keysteps	Opt. Keysteps	Proc. Mistakes	Miss. Keysteps	Fut. Keysteps
-	Uniform Baseline	-	50.00	59.72	50.00	53.76	60.65
Instance-Level	Graph-Based	Ground Truth	**82.34**	**71.30**	**61.87**	**72.99**	**83.77**
Instance-Level	End-to-End	Ground Truth	<u>58.68</u>	60.28	<u>61.31</u>	<u>58.33</u>	<u>73.73</u>
Procedure-Level	Graph-Based	Keystep Assignment	50.80	51.22	55.60	50.58	65.20
Procedure-Level	End-to-End	Keystep Assignment	55.43	<u>66.17</u>	59.28	57.96	71.25

Table 17. Results for the procedure understanding task. Best results are reported in bold, the second best results are underlined. All results are in percentage.

video segment features extracted with EgoVLPv2 to three dedicated MLPs. Figure 41(b) illustrates the architecture of the baseline. At training time, MLPs are supervised from the pseudo-labels obtained by graph-based baselines using Mean Squared Error (MSE) score to align the predicted probability distributions to the supervising ones. Missing keysteps and procedural mistakes are predicted from the outputs of the MLP components as in graph-based baselines.

Results We carried out experiments on the following four scenarios: *Install a Wheel, Remove a Wheel, Clean and Lubricate the Chain, First AID - CPR, COVID-19 rapid antigen test*. We evaluated our baselines using the calibrated Average Precision (cAP) [30]. Note that, according to this measure, a random baseline would on average achieve a performance of 50%.

Table 17 reports the results obtained by our baselines and compares them against those produced by a "uniform" baseline, predicting previous/optional/mistakes/missing/next keysteps with equal probabilities. Results show that the graph-based baseline relying on ground truth annotations significantly outperforms the uniform baseline for some of the tasks, such as previous and missing keystep predictions. This suggests that even simple keystep co-occurrences are informative to some degree of the overall structure of the procedure. The limited performance gains on optional keystep prediction, procedural mistake detection and future keystep prediction highlight the complexity of the task and the need for further research. The end-to-end model trained with instance-level supervision achieves lower or similar performance, trading accuracy for test-time efficiency, due to the absence of an explicit graph. Procedure-level baselines achieve lower performance due to the limited level of supervision they can rely on. The only exception is for the optional keysteps where the end-to-end baseline outperforms the others methods.

Contributions statement Antonino Furnari led the procedure understanding benchmark, and contributed to the task definition, the annotation guidelines, the baseline design, and paper writing. Giovanni Maria Farinella contributed to the task definition, the annotation guidelines, the baseline design, and paper writing. Luigi Seminara contributed to the annotation guidelines, the baseline design, and paper writing; he also developed tools for data annotation, and the baselines for the benchmark. Francesco Ragusa contributed to the annotation guidelines, the baseline design, and paper writing; he also developed tools for data annotation, and the baselines for the benchmark. Kumar Ashutosh contributed to the annotation guidelines, baseline design, and development of data annotation tools. Michael Wray contributed to the task definition, the annotation guidelines, the baseline design, and paper writing. Siddhant Bansal contributed to the task definition, the annotation guidelines, the baseline design, and paper writing. Gene Byrne contributed to the task definition, the annotation guidelines and the baseline design. Tushar Nagarajan contributed to the task definition, and the baseline design. Lorenzo Torresani contributed to editing this section.

13.C. Ego-(Exo) Proficiency Estimation

Annotations The proficiency estimation benchmark consists of two task variants: (1) *demonstrator proficiency estimation*, where the goal is to estimate the absolute skill level of a participant at the task, and (2) *demonstration proficiency estimation*, where the goal is to perform fine-grained analysis of a given task execution to identify good actions from the participant and suggest tips for improvement. We now provide a detailed description of the annotations used in each of these task variants.

Demonstrator proficiency estimation. In this task variant, we assign proficiency labels to each person performing activity demonstrations (one label per person). The proficiency labels span four distinct categories: novice, early expert, intermediate expert, or late expert. We find four proficiency classes makes the task challenging but still approachable. Subtle variations between 5 or more levels of proficiency can be insufficiently observable from vision alone, and even difficult for expert annotators to reach consensus. Most levels correspond to experts since Ego-Exo4D videos are dominantly targeted towards expert participants who can perform the task successfully (see Sec. 10).

We derive annotations for this task from participant surveys (see Sec. 11.B) and expert commentary (see Sec. 12.A). Participant surveys contain responses to questions about prior experiences in the task such as "How many years have you been doing this task?", and "Do you

	Demonstrator proficiency			Demonstration proficiency		
	Train	Val	Test	Train	Val	Test
Basketball	576	105	141	146	47	19
Bike repair	-	-	-	41	9	15
Cooking	83	20	36	80	24	39
Dance	408	124	144	80	35	27
Health	-	-	-	42	12	16
Music	149	39	43	94	32	35
Rock Climbing	620	162	229	65	15	22
Soccer	68	16	24	8	3	6
Total	1904	466	617	556	177	179

Table 18. Distribution over video takes in proficiency estimation benchmark.

have any qualifications/professional training related to the task?" (see Tab. 3 for the complete list). On the other hand, expert commentary is performed by task-specific experts and includes 1 to 10 proficiency scores for each video from the participant (see Sec. 12.A). After consulting with experts hired for each scenario, we designed scenario-specific conversion functions that use the surveys and expert commentaries to produce an estimate of a participant's proficiency score (see Tab. 19). For example, in basketball and soccer, we use the years of experience to determine skill level since we found this to be an accurate indicator of skill based on analyzing the videos. On the other hand, to determine skill level in bouldering, we use the highest difficulty level of the route solved by the participant.

Note that we exclude the bike repair and health scenarios from the demonstrator proficiency task for different reasons. The distribution of participants for bike repair is heavily skewed towards late expert participants, which would heavily bias the task of demonstrator proficiency. The predominant activity in the health collection is COVID testing, where skill levels are hard to determine due to the simplicity of the task.

See Fig. 42 for a distribution over proficiency scores within each scenario. We split our dataset into train/val/test splits based on the common split shared across benchmarks. The dataset statistics are shown in Tab. 18.

Demonstration proficiency estimation. For this variant of the task, we leverage temporally localized annotations that include the timestamps of steps demonstrated in the video as well as the proficiency category for each demonstrated step instance (i.e., good execution or a mistake). For this task, we consider all 8 scenarios, as shown in Tab. 18. We derive annotations for this task from expert commentary, where task experts carefully analyze videos and provide timestamped commentary on the participant's performance (see Sec. 12.A). In particular, given a single timestamped comment from an expert, we annotate whether the comment describes a good execution and/or provides tips for

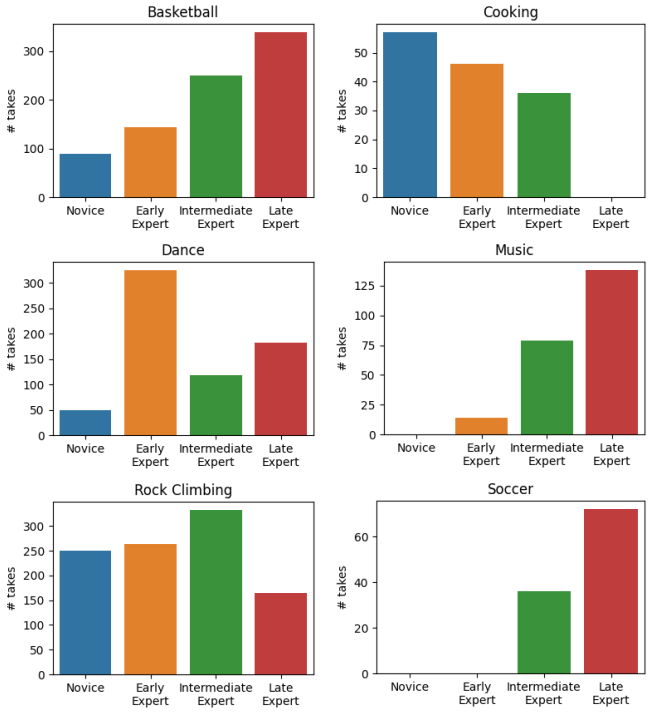


Figure 42. Distribution of demonstrator proficiency scores per scenario.

improving the participant's skill level. See Tab. 20 for example annotations. These annotations are then associated with the timestamp provided with each comment to obtain a list of timestamps for good executions $\{t_1^g, t_2^g, \cdots\}$ and tips for improvement $\{t_1^i, t_2^i, \cdots\}$ in each video. The objective then is to train models that can predict timestamps of good executions or tips for improvement. This amounts to a temporal action localization task for the actions "good executions" and "tips for improvement". While it would be interesting to build systems to further elaborate what these are (e.g., a specific type of tip, or a natural language explanation), we reserve this for future work. Overall, the demonstration proficiency estimation task consists of 556 train / 177 val / 179 test videos (see Tab. 18 for a breakdown per scenarios).

Formal task definition We now formally define the two problem formulations. In both cases, we are given as input an egocentric video and a set of M exocentric videos of a demonstrator performing a task: $\mathcal{V} = \left(\mathcal{V}_{ego}, \mathcal{V}_{exo}^{1-M}\right)$. The ego and exo views are time-aligned.

Demonstrator proficiency estimation. The goal here is to estimate the demonstrator's skill level from one or more task demonstrations. We wish to learn a function f that can estimate the demonstrator's proficiency, i.e.,

Scenario	Novice	Early Expert	Intermediate Expert	Late Expert
Basketball	$X \in [0, 1)$	$X \in [1, 3)$	$X \in [3, 10)$	$X \geq 10$
Soccer	$X \in [0, 1)$	$X \in [1, 3)$	$X \in [3, 10)$	$X \geq 10$
Dancing	$X \in [0, 3)$	$X \in [3, 5)$	$(X \in [5, 10)) \vee ((X \geq 10) \wedge \neg P)$	$(X \geq 10) \wedge T$
Bouldering	$H \leq$ V3	$H ==$ V4	$H ==$ V5	$H \geq$ V6
Music (violin)	$(X \in [0, 3)) \vee (N \in [0, 500))$	$(X \in [3, 5)) \vee (N \in [500, 1000))$	$(X \in [5, 10)) \vee (N \in [1000, 10000))$	$(X \geq 10) \vee (N \geq 10000)$
Music (guitar)	$(X \in [0, 1)) \vee (N \in [0, 500))$	$(X \in [1, 3)) \vee (N \in [500, 1000))$	$(X \in [3, 10)) \vee (N \in [1000, 10000))$	$(X \geq 10) \vee (N \geq 10000)$
Music (piano)	$(X \in [0, 1)) \vee (N \in [0, 500))$	$(X \in [1, 5)) \vee (N \in [500, 1000))$	$(X \in [5, 10)) \vee (N \in [1000, 10000))$	$(X \geq 10) \vee (N \geq 10000)$
Cooking	$P < 3.5$	$P \in [3.5, 5)$	$P \in [5, 8)$	$P \geq 8$

Table 19. **Annotations for demonstrator proficiency estimation.** We designed scenario-specific conversion functions that take in participant surveys and expert commentary assessments to estimate proficiency of participants (i.e., novice, early expert, intermediate expert, and late expert). Legend: X = years of experience performing the task, T = professional training in the task, H = highest difficulty level solved by participant in bouldering, N = estimated number of times performing the task, P = average proficiency rating from expert commentary.

Scenario	Expert comment	Good execution	Tips to improve
Basketball	Nice release. I like the follow through here. You'd like to see the guide hand maybe up a little bit higher on the release of that shot. Maybe to give it better ball control when you're letting go of the shot.	Yes	Yes
Basketball	Great footwork, left foot take off, lifting of the right knee and extending that body up. Love how he's looking up, checking out the backboard, shooting hand behind the basketball. Nice job.	Yes	No
Basketball	He's also really far away from his body and the more he can keep his arm up by his ear, it will give him the most opportunity to make the basket without the defense interrupting.	No	Yes
Bike repair	It's a great method to always double check or do a pre-check before beginning work on a bicycle to make sure the issue that you are working to fix is the only issue that is occurring. If not, you could find a secondary issue or something else that may be greater than the one you are currently working on.	Yes	No
Bike repair	As you can see she clearly slipped on loosening the nut which essentially creates damage to the surface of the nut itself and can round out the nut.	No	Yes
Bouldering	The climber was efficiently able to position herself with one hand on each hold at the start and had, once her hands were positioned, she matched her feet on the hold and efficiently moved to the next hold.	Yes	No
Bouldering	And since she popped out and is swinging out, she can't really keep the tension through her one arm because she's so locked off. So it caused her to kind of just fall off the wall and lose all tension throughout all of her body.	No	Yes
Cooking	You can see there, she's not able to stir properly. She has to push it around, which means that the lime is not gonna be very evenly distributed among the pieces of tomato and cucumber.	No	Yes
Cooking	Using a grinder for fresh pepper is an excellent way to get a lot of flavor. The fresh grind of pepper as opposed to buying already ground pepper really expels the oils and everything in those peppercorns and allows the flavor to be as big as it can possibly be.	Yes	No

Table 20. **Annotations for demonstration proficiency estimation.** We annotated expert comments about a participant's task execution with tags indicating whether each comment describes a good execution or suggests tips for improving skills. Note that the same comment might describe one aspect of the task as being good while suggesting improvements in another aspect (e.g., see row 1).

$f(\mathcal{V}) = y^{prof} \in \{$novice, early expert, intermediate expert, late expert$\}$.

Demonstration proficiency estimation. On the other hand, the goal in this task variant is to identify parts of the video where the execution was good or needs further improve-

ment. Thus, this time the objective is to learn a function h that can temporally localize instances of good executions and instances of demonstrations that require improvement. Formally, we can express function h as:

$$\hat{G}, \hat{I} = h(\mathcal{V}), \tag{2}$$

where $\hat{G} = \{t^g_1, t^g_2, \cdots, t^g_{|G|}\}$ are the timestamps where the participant shows good task execution, and $\hat{I} = \{t^i_1, t^i_2, \cdots, t^i_{|I|}\}$ are the timestamps where the participant needs to improve their skill level.

These tasks inherently benefit from multi-view data. Egocentric video captures fine-grained information about the demonstrator's hand pose and object interactions, which can be critical to estimating proficiency in tasks that require fine-grained interactions such as cooking (e.g., chopping vegetables) and music (e.g., placement of fingers on the guitar). On the other hand, the exocentric videos provide broader information about the demonstrator's body pose, which can be highly indicative of proficiency in tasks that require extensive physical motion such as basketball, soccer, and dancing. Our EgoExo4D dataset has 5 views (1 egocentric view, and $M = 4$ exocentric views). We run the proficiency tasks in two settings: one where the exo view is available at test time, and one where it is not. For the latter, benchmarking baseline models with only the egocentric view is important when the target is augmented reality and mobile robotics applications. For the former, benchmarking with both egocentric and exocentric views is informative to capture the multi-view nature of the problem.

Note that the input to the model excludes textual descriptions/narrations of the activity, audio, gaze sensor readings, and any subject information, which would simplify the task significantly at the expense of general applicability since these signals are typically not available for in-the-wild video. We believe the formulation proposed here will encourage the development of video-based methods that (1) do not rely on explicit subject information such as gender, age, ethnicity, etc., and (2) learn to estimate proficiency based on visual cues rather than high-level textual activity descriptions or alternate modalities like sound and eye gaze.

Baselines Next, we define our proposed baselines for each task in the benchmark.

Demonstrator proficiency estimation. We approach this task as a classification problem with four proficiency classes: novice, early expert, intermediate expert, and late expert. We adopt the TimeSformer [13] model for our experiments. TimeSFormer is a video transformer designed for video action recognition/classification that introduces a novel decoupled spatiotemporal attention mechanism. We train one model on the egocentric view ("ego model"), and a separate model on all 4 exocentric views ("exocentric model"). We resize all videos to 448 pixels along the smallest dimension and use a clip size of 16 frames with a frame rate of 16 FPS. The models are trained to classify individual clips on 8 Quadro RTX 6000 GPUs for 15 epochs. We use the cross-entropy loss as our training objective. At inference time, we employ a late fusion strategy to incorporate information from both egocentric and exocentric video streams. We average the softmax predictions across both egocentric and exocentric models to obtain the final video label prediction. We also average results over three spatial crops during inference following prior work [13]. We report the top-1 accuracy metric.

Demonstration proficiency estimation. We treat this task as a temporal action localization problem with two action categories: 'good execution' and 'needs improvement'. We adopt the ActionFormer [190] model for our experiments. Unlike traditional action localization that defines time windows as outputs, we instead perform timestamp regression since our annotations contain only a single point in time for each good execution or tip for improvement. We accordingly adapt ActionFormer's post-processing strategy and evaluation metrics. In our task, the predicted timestamps correspond to frames retained after non-maximum suppression (NMS). We remove the regression head of ActionFormer and infer the predicted timestamps from the indices of frames retained after NMS. We also modify the NMS module of ActionFormer to rely on the L_1-distance between predicted timestamps instead of the tIoU between segments used in [190]. During training, we keep the classification loss from [190] and replace the regression loss with the loss function defined in [75].

We also modify the evaluation setup to use the L_1-distance between the predicted and ground-truth timestamps instead of tIoU between segments. Therefore, we have metrics thresholded by L_1-distance in seconds as opposed to tIoU values. We train our models with Omnivore features [45] extracted with a clip size of 32 frames and a stride of 16 frames from all the videos. For the experiments involving multiple views (i.e., multiple exo views or ego + exo views), we simply concatenate the features for all views at each time step.

Results

Demonstrator proficiency estimation.

We present results for demonstrator proficiency estimation in Tab. 21. We include two naïve baselines to account for biases in the dataset. The random baseline uniformly samples one skill level at random. The majority-class baseline predicts the majority class within each scenario. While the random baseline performs poorly, the

Method	Pretraining	Val accuracy			Test accuracy		
		Ego	Exos	Ego + Exos	Ego	Exos	Ego + Exos
Random	-	24.9	24.9	24.9	24.9	24.9	24.9
Majority-class	-	31.1	31.1	31.1	35.3	35.3	35.3
TimeSFormer [13]	-	42.3	40.1	**40.8**	42.3	48.6	47.8
TimeSFormer [13]	K400	42.9	39.1	38.6	47.3	**49.4**	**48.5**
TimeSFormer [13]	HowTo100M	**46.8**	38.2	39.7	48.5	38.1	40.7
TimeSFormer [13]	EgoVLP	44.4	**40.6**	39.5	**48.6**	46.0	47.2
TimeSFormer [13]	EgoVLPv2	45.9	38.0	37.8	39.5	46.7	45.7
Inference with multiple takes per demonstrator							
TimeSFormer [13]	K400	43.3	38.0	38.6	42.1	52.4	52.0

Table 21. **Demonstrator proficiency estimation benchmark.** We report top-1 accuracies for various baselines on the demonstrator proficiency estimation task.

majority-class baseline achieves higher accuracies, which results from scenario-specific biases in the distribution of expertise levels (see Fig. 42). TimeSFormer trained from random initialization outperforms the naïve baselines by a significant margin, demonstrating the ability of learned methods to quantify skill levels from videos. In both val and test splits, the ego videos are sufficient to achieve good performance. In the test split, using the exo videos further improves performance in tasks such as bouldering, highlighting the complementary nature of the ego and exo viewpoints. When models are initialized using HowTo100M, EgoVLP and EgoVLPv2 pre-training, we observe good improvements using the ego videos, while the performance using exo videos does not improve over random initialization. In contrast, K400 initialization leads to significant improvements across multiple settings. In all cases, we found that fusing the predictions from the ego view and exo views does not lead to improved results, likely due to the simplicity of our late-fusion strategy. In the last row of Tab. 21, we further report results when evaluating the model from row 5 on all demonstrations from the same participant (as opposed to a single demonstration). This multi-take analysis benefits the exo model and the ego + exo model fusion on the test split, but fails to improve on other cases, likely due to the simplicity of our late-fusion across multiple takes. Nevertheless, this highlights the potential for developing more complex strategies to study a participant's skill level across several demonstrations to obtain more accurate skill estimates. We show scenario-specific results in Tab. 22. The TimeSFormer model achieves good performance with the egocentric view in cooking since a close-up view of the objects of interest and hand poses is essential to assessing skill in these scenarios. On the other hand, the model performs better with the exocentric view in music and basketball since the overall body pose is a useful indicator of proficiency. Unfortunately, the TimeSFormer model fails to improve over the majority-class baseline for soccer scenarios. Overall, our benchmark presents new challenges for video-based skill understanding and our results highlight the difficulty of the task, suggesting good scope for improvement in future work.

Demonstration proficiency estimation. We present results for the demonstration proficiency estimation task in Tab. 23. We include three naïve baselines along with ActionFormer [190]. The "Random tips/good exec." baseline randomly predicts a tip or a good execution label every 5.97 seconds, i.e., the average temporal span between adjacent annotations in our dataset. The "Uniform tips" baseline predicts a tip for improvement label every 5.97 seconds. The "uniform good exec." baseline predicts a good execution label every 5.97 seconds. We evaluate ActionFormer models trained on ego only, exo only and ego + exo views. All naïve baselines perform poorly on this task. The learned ActionFormer baseline outperforms the naïve baselines by a good margin. However, the absolute mAP scores are fairly low, suggesting that the task is very challenging and has a significant scope for improvement in methods.

Contribution Statement Santhosh Kumar Ramakrishnan co-led the proficiency estimation benchmark, co-developed the task formulation, and advised the baseline development. Gedas Bertasius co-led the proficiency estimation benchmark, co-developed the task formulation, and advised the baseline development. Arjun Somayazulu developed the demonstrator proficiency estimation baselines. Abrham Gebreselasie developed the demonstration proficiency estimation baselines. Maria Escobar contributed to the task definition and the baseline design. Eugene Byrne contributed to the task definition and the baseline design. Miguel Martin developed an interface for obtaining proficiency estimation scores from the recruited experts. Suyog Jain contributed to an annotation pipeline for demonstration proficiency estimation. Devansh Kukreja built the render flow to generate

Val results

Scenario	Majority-class -	TimeSFormer [13] Ego	Exos	Ego + Exos
Basketball	36.19	51.43	52.38	**55.24**
Cooking	**50.00**	45.00	35.00	35.00
Dancing	51.61	**55.65**	42.74	42.74
Music	58.97	46.15	**69.23**	56.41
Bouldering	0.00	**25.31**	17.28	17.28
Soccer	62.50	56.25	**75.00**	**75.00**

Test results

Scenario	Majority-class -	TimeSFormer [13] Ego	Exos	Ego + Exos
Basketball	38.30	56.74	**63.83**	62.41
Cooking	36.11	**52.78**	33.33	33.33
Dancing	45.14	**46.53**	43.06	41.0
Music	60.47	55.81	**76.74**	74.42
Bouldering	19.65	39.30	**42.80**	**42.80**
Soccer	**62.50**	50.00	41.67	41.67

Table 22. **Breakdown of results for demonstrator proficiency estimation across scenarios.** We report top-1 accuracies per scenario for the TimeSFormer model with K400 pre-training.

frame-aligned videos of each camera for each take as model input. Kristen Grauman contributed to the task formulation. Lorenzo Torresani contributed to editing this section.

13.D. Ego Pose

This family of tasks is motivated by recovering the skilled movements of experts in the extreme setting of monocular ego-video input.

13.D.1 3D human body pose from ego-video

Annotations The 3D human body pose annotation process consists of two main stages: (1) automatic ground truth generation, and (2) manual multi-view keypoint annotation/correction. Through this process we derive 3D keypoint annotations for approximately 14M frames.

In the automatic ground truth generation phase, we use off-the-shelf models [25] to predict the 2D bounding boxes from each of the exocentric views. Since there could be multiple people in the scene and we only want to consider the one wearing the egocentric camera, we project the 3D headset location from the MPS output to select which box corresponds to the camera wearer. Then, we run an off-the-shelf 2D human keypoint detector [25] for each bounding box to obtain the 2D keypoints. Finally, we run 3D triangulation with RANSAC to minimize the reprojection errors to obtain the 3D keypoints for the camera wearer.

In the manual annotation phase, we import the undistorted frames and the reprojected 2D keypoints into our multi-view annotation interface.

Ego-Exo4D offers the largest available manually annotated body pose (376K 3D/2M 2D) and hand pose (68K 3D/340K 2D) annotations. Along with this, we also provide 9.2M/47M (body) and 4.3M/21M (hand) automatically generated groundtruth 3D and 2D poses, totaling about 13.M frames In total, we have approximately 14M frames of 3D ground truth (GT) and pseudo-GT combined across body and hands. To our knowledge, this represents the largest collection of body pose annotations in the literature, whether for ego or exo video.

How good is the auto GT? Between manual and automatic annotations, the body and hand MPJPEs are 3.33 cm and 1.87 cm, respectively, much smaller than the best baseline methods. It is important to note that Ego-Exo4D tackles real-world scenarios with five or fewer cameras rather than controlled environments. This introduces challenges like increased occlusions from body and objects along with limited view of hands from distant cameras. Despite this, our auto generation pipeline surpasses baselines, showcasing robustness and efficacy. Experiments below further show performance boosts across baselines when using automatic ground truth, demonstrating its effectiveness. Note that automatic GT and manual GT are not mutually exclusive, and people can choose whether/how automatic GT is used for training.

Formal task definition In our ego-pose task, the goal is to estimate the 3D human pose using either an RGB video input sequence $\mathcal{V}_{ego}$, an IMU sensor sequence $\mathcal{T}_{imu}$, or both. For a current frame t, given either an egocentric video $\mathcal{V}_{ego} = \{\mathcal{V}^k, ..., \mathcal{V}^{t-1}\}$ or an IMU sequence $\mathcal{T}_{imu} = \{\mathcal{T}^k, ..., \mathcal{T}^{t-1}\}$, where k is a time window in the past, the goal is to predict the human pose P at time t, where $P^t \in \mathcal{R}^{17\times3}$ for the 17 annotated joints. Note that at test time, we only estimate the error across the visible annotated joints at frame t. Note that, since the EgoPose benchmark is aimed at promoting the development of methods that perform body pose estimation solely from first-person raw video or IMU data, the input *excludes* egocentric modalities that would unfairly simplify the task (e.g., audio captured from a wearable camera, eye gaze), as well as exocentric video or any signals that can be extracted from it.

Metrics To evaluate the performance of body pose estimation approaches we calculate the Mean Per Joint Position Error (MPJPE) in centimeters (cm), and the Mean Per Joint Velocity Error (MPJVE) in meters per second (m/s).

Baselines We evaluate several baselines methods. Kinpoly [98] is a physics-based simulator that estimates pose using hard physics constraints applied to an underlying artificial humanoid model. EgoEgo [80] is a diffusion based

Val results

Method	Ego				Exos				Ego + Exos			
	$mAP_{0.25}$	$mAP_{0.5}$	$mAP_{1.0}$	Avg.	$mAP_{0.25}$	$mAP_{0.5}$	$mAP_{1.0}$	Avg.	$mAP_{0.25}$	$mAP_{0.5}$	$mAP_{1.0}$	Avg.
Random tips/good exec.	0.47	1.45	4.70	2.21	0.47	1.45	4.70	2.21	0.47	1.45	4.70	2.21
Uniform tips	0.50	1.66	5.31	2.49	0.50	1.66	5.31	2.49	0.50	1.66	5.31	2.49
Uniform good exec.	0.45	1.63	4.87	2.32	0.45	1.63	4.87	2.32	0.45	1.63	4.87	2.32
ActionFormer [190]	**1.57**	**4.25**	**9.60**	**5.14**	**1.19**	**3.10**	**6.99**	**3.76**	**0.96**	**2.61**	**6.72**	**3.43**

Test results

Method	Ego				Exos				Ego + Exos			
	$mAP_{0.25}$	$mAP_{0.5}$	$mAP_{1.0}$	Avg.	$mAP_{0.25}$	$mAP_{0.5}$	$mAP_{1.0}$	Avg.	$mAP_{0.25}$	$mAP_{0.5}$	$mAP_{1.0}$	Avg.
Random tips/good exec.	0.49	1.67	5.23	2.47	0.49	1.67	5.23	2.47	0.49	1.67	5.23	2.47
Uniform tips	0.44	1.52	5.08	2.35	0.44	1.52	5.08	2.35	0.44	1.52	5.08	2.35
Uniform good exec.	0.45	1.38	4.50	2.11	0.45	1.38	4.50	2.11	0.45	1.38	4.50	2.11
ActionFormer [190]	**1.08**	**2.97**	**7.20**	**3.75**	**0.91**	**2.43**	**6.08**	**3.14**	**0.91**	**2.71**	**6.77**	**3.47**

Table 23. **Demonstration proficiency estimation benchmark.** We report the mean average precision (%) for various baselines on the demonstration proficiency estimation task. mAP_k is measured at an L_1-distance threshold of k seconds. The average mAP (Avg.) measures the mAP averaged across $k = \{0.25, 0.5, 1.0\}$ seconds.

method which maps a sequence of 6 DoF head estimates to a full sequence of 3D body joint estimates. We also develop a method that combines current IMU-based body pose estimation models Bodiffusion [18] and AvatarPoser [64]. Moreover, to gauge the performance of deep-learning-based methods, we implemented a more straightforward approach. We create a static pose baseline, which consists of fixing the 3D human body pose prediction to be the average pose in the training set and translating it according to the IMU sensor. Thus, the fixed prediction matches the camera location at each frame.

- **Kinpoly.** Kinpoly [98] proposes to use a simulated humanoid to track head pose and create full-body motion based on action types. Based on the input head-pose and action type (out of push, step, avoid, sit, and locomotion), Kinpoly synthesizes realistic human pose and human-object interactions inside a physics simulator. Different from kinematic-based methods that directly output joint angles or positions for pose estimation, Kinpoly outputs joint torques as the final product and controls a simulated humanoid for pose estimation. The laws of physics and human dynamics regulates the output pose to remove artifacts such as jitter and penetration.
- **EgoEgo.** EgoEgo [80] uses a two-step approach for egocentric body pose estimation, by estimating the head pose from the egocentric video first, and then using a diffusion model to generate the full body motion sequence based on the head pose sequence. For head pose estimation, it obtains the initial head pose trajectory using DROID-SLAM [160], and then uses learning-based methods to correct the head pose, including a GravityNet to estimate the additional rotation and a HeadNet with optical flow features as input to estimate the scaling factor to the trajectory. The full body pose is generated with a modified version of DDPM [53] that is conditioned on head pose and trained on AMASS [99]. We show the evaluation of the conditional diffusion part here.
- **IMU-based.** This baseline is inspired by state-of-the-art methods that use transformer-based models for body pose estimation from sparse inputs[18, 64]. We adapt these methods to utilize 3D positions as opposed to the traditional parametric body model. During the training phase, the model was subjected to $40,000$ iterations, using the Adam optimizer with a learning rate of $1e^{-4}$. The window size for temporal analysis was set at 40 frames, and we minimized the Mean Squared Error (MSE) loss between predicted poses and ground truths. As for the input, our model receives a sequence of head poses captured by the device.

Results Table 24 shows the evaluation results of all the baseline approaches. First, note that the static pose baseline obtains a significantly higher MPJPE than all the other approaches. This finding suggests that the poses across different scenarios in the dataset are extremely diverse. Thus, attempting to have the same static pose for all test cases is unfeasible. In contrast, the proposed baseline implementations achieve notable enhancements in performance. Table 25 shows the performance of each method per scenario. Although most scenarios have comparable results, activities like bouldering have a higher error since they require a translation along the vertical axis, which differs from a typical movement in other scenarios. While these developments are promising, we believe that further refinement is possible, especially in lower body pose estimation and to ensure temporal consistency in predictions.

Method	Validation		Test	
	MPJPE	MPJVE	MPJPE	MPJVE
Static pose	163.94	-	150.95	-
EgoEgo [80]	24.35	0.71	28.78	0.57
Kinpoly [98]	22.66	0.74	25.80	0.60
IMU-based [18, 64]	20.86	1.87	19.88	1.70

Table 24. **Results for the 3D human body pose benchmark.** We report the Mean Per Joint Position Error in cm and the Mean Per Joint Velocity Error in m/s for all the baseline approaches.

Scenario	EgoEgo	Kinpoly	IMU-based
Basketball	20.00	30.05	20.25
Soccer	24.34	18.72	20.83
Bike repair	29.07	24.57	17.49
Cooking	24.97	23.26	14.62
Health	34.13	30.47	9.82
Bouldering	42.15	40.34	25.87
Dance	28.71	25.42	22.71
Music	33.15	34.59	19.41

Table 25. **Body pose estimation Test results per scenario.** We report the Mean Per Joint Position Error in cm for the Test split.

13.D.2 3D human hand pose from ego-video

Annotations As for the case of body pose annotation, the 3D human hand pose annotation process also consists of two stages, i.e., the automatic ground truth generation and the manual multi-view keypoint annotation. Compared to the body pose, the main difference in automatic ground truth generation is that we also detect hand keypoints from the egocentric frame, and we use the result from the whole body pose estimation to infer the hand locations when there are multiple people in the scene. Similarly, for manual annotation, besides the exocentric frames, we also show the annotators the egocentric frames to allow them to annotate/correct hand keypoints.

For each annotated joint in manual annotations, we provide the number of views used for triangulation as the indicator of the confidence for the provided ground truth data. Meanwhile, the correction the annotators make for hand joints on ego images can serve as the indicator to understand the difficulty for hand reconstruction from the given ego view.

Formal task definition The ego hand pose task entails predicting the three-dimensional coordinates of the camera wearer's hands from the egocentric frames. Frames from the ego view are extracted and undistorted for both training and evaluation. For baselines requiring 2D hand bounding boxes, we project 3D hand joints in camera coordinates to 2D image planes using the provided intrinsic matrix for the crops.

Note that the inputs of the task do not include depth maps, any additional views from ego or exo cameras, camera pose information, and IMU or active range sensor measurements. We explicitly *exclude* such information for this benchmark to promote the applicability of the methods to general hand pose estimation problems using monocular images.

Metrics To cope with methods estimating wrist-origin and camera-origin hand pose, the ego hand pose baselines are evaluated according to both the MPJPE and the PA-MPJPE metrics. The MPJPE is camera-relative while the PA-MPJPE calculates the average 3D joint distance after performing Procrustes Alignment on wrist-origin hand poses. Both metrics are reported in millimeter (mm) unit.

Baselines We implemented and trained several baseline models for ego hand pose estimation. To estimate the 3D hand joint from monocular 2D ego view images, 2D heatmaps can be explicitly estimated and lifted to 3D space, or 3D joints can be directly estimated from extracted 2D features. The feature extractor backbone could be CNN-based or transformer-based. The proposed baseline methods cover different choices of model designs. All the baseline models work on single frame images without temporal information. The baseline models are trained on manual or manual+automatic annotations, and are only evaluated on manual annotations.

Notably, most baseline methods generate hand mesh as final results in their original paper. We modified them to be trained and supervised only on 2D/3D hand joints (not on hand mesh) to fit the benchmark.

- **THOR-net.** THOR-net [1] uses Keypoint-RCNN as the feature extractor to obtain 2D information and derive 2D hand keypoints heatmaps explicitly. The method then lifts 2D estimates to the 3D space using GraFormer [197], which is a model consisting of Graph Convolutional layers and Attention layers. We use only the 2D-to-3D pose GraFormer branch in THOR-net to adapt the method to our task. The training takes around 4 hours for the manual dataset on a GeForce RTX 4090 Graphics Card, and around 10 hours for the dataset combining manual and automatic annotations.
- **HandOccNet.** HandOccNet [112] uses a ResNet50[51]-based FPN [88] to extract 2D features. The method then uses two Transformer-based modules: Feature Injecting Transformer (FIT) to inject hand information into occluded region, and Self-Enhancing Transformer (SET) to further refine the 2D features. The method proposes a regressor based architecture to produce 2D keypoints,

	Manual		Manual+Auto	
	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
METRO* [84]	-	21.54	-	21.54
THOR-net [1]	51.06	17.88	50.10	16.34
HandOccNet [112]	59.30	18.70	59.00	17.90
POTTER [200]	32.71	11.74	32.02	11.62

Table 26. MPJPE and PA-MPJPE in mm for ego hand pose baseline models. * denotes methods *not* trained on the benchmark.

	THOR-net[1]	HandOccNet[112]	POTTER[200]
Params (M)	59.5	37.22	14.5
MACs (G)	123.6	15.5	5.2

Table 27. Number of parameters and MACS for the different ego hand pose baselines.

MANO [138] pose, and MANO shape parameters to predict joints and vertices. To accommodate our baseline, only 2D keypoints and 3D joints location losses are used in the training phase. The training takes around 2 hours for the manual dataset on 8 NVIDIA V100 Graphics Cards.

- **POTTER.** POTTER [200] proposes Pooling Attention Transformer (PAT) to extract 2D visual features, which significantly reduces the memory and computational cost without sacrificing performances. The method then applies a mesh regression head HybrIK [79] to generate 3D joint and mesh results. The training takes around 43 minutes for manual dataset, and around 4 hours for manual+auto dataset on a GeForce RTX 4090 Graphics Card.
- **METRO.** METRO [84] extracts a CNN-based global image features. The method then uses a transformer encoder to jointly model vertex-vertex and vertex-joint interactions, and outputs 3D joint coordinates and mesh vertices simultaneously. Since the training of METRO strongly depends on hand mesh supervision, which is not present in the annotations, we borrowed the checkpoint trained on FreiHand [72] dataset and run the inference only, without training it on our benchmark.

Results We report the MPJPE and PA-MPJPE of the baseline models in Table 26, and their corresponding parameter numbers and multiply-accumulate operations (MACs) in Table 27. We further analyze the error distribution across different hand joints. Figure 43 shows that the thumb finger and finger tips tend to have larger errors, mostl likely because they are occluded or invisible more often.

For each annotated joint, the manual annotations keep record of the number of views where the joint is visible. The visible 2D observation is then used for triangulation in 3D ground truth generation. This can be taken as an indicator of the uncertainty of the ground truth, and the difficulty level for the estimation of the joint (usually, a joint visible by fewer views indicates that it is more entangled with objects or other part of the hand). Table 28 shows that the PA-MPJPE decreases as the visible number of views increases. To guarantee the ground truth accuracy, all experiments are performed only on joints at least visible in 3 cameras.

# visible views	3	4	5	6
PA-MPJPE (mm)	14.01	12.15	11.03	10.02

Table 28. PA-MPJPE for joints that are visible in different number of views (including ego and exo views). Results generated from POTTER [200] evaluation.

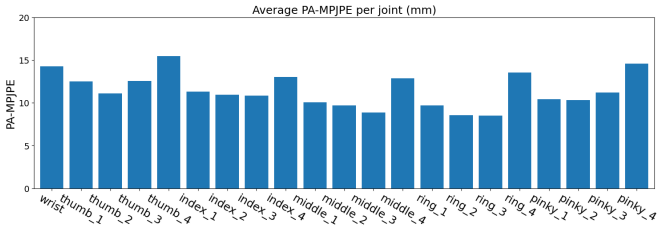


Figure 43. Average PA-MPJPE for each joint. Results generated from POTTER [200] evaluation.

Contribution Statement Kris Kitani co-led the ego-pose benchmark, and provided directional guidance on the task definition, the annotation methodology, and the baseline development. Jianbo Shi co-led the ego-pose benchmark, and provided directional guidance on automatic 3D hand pose generation and development of ego hand pose baseline methods. Maria Escobar led the ego-pose body baseline development, implemented the IMU-based baseline and contributed to experiment analysis. Cristhian Forigua developed the static pose baseline and contributed to the implementation of the IMU-based baseline. Fu-Jen Chu developed the multi-view annotation UI, the hand pose annotation guidelines, and the data preprocessing code for ego-pose annotation; he also trained and evaluated the HandOccNet baseline. Rawal Khirodkar developed the multi-view triangulation and 3D body keypoint estimation pipeline. Zhengyi Luo contributed to the Kinpoly baseline and to the coordinate transform for Aria head poses. Shan Su led the ego-pose hands baseline development, and contributed to automatic 3D hand pose generation, task definition, and annotation development; she also evaluated the baseline using METRO. Suyog Jain developed the annotation pipeline to scale the annotation collection, worked on training the annotators and managed the overall annotation process. Miguel Martin contributed to the automatic ground truth generation pipeline and provided high-level coordination of the body and hands automatic ground truth

generation. Jinxu Zhang developed automatic ground truth generation for 3D hand pose; he also trained and evaluated baseline model POTTER. Yiming Huang trained and evaluated the baseline model THOR-net. Zhifan Zhu developed the METRO hand pose baseline method. Jing Huang led the automatic ground truth generation effort, refined body pose annotation guidelines, coordinated ego-pose body baseline development, and ran the EgoEgo body pose baseline. Lorenzo Torresani contributed to editing this section.