EgoDex: Learning Dexterous Manipulation from Large-Scale Egocentric Video

Author Names Omitted for Anonymous Review

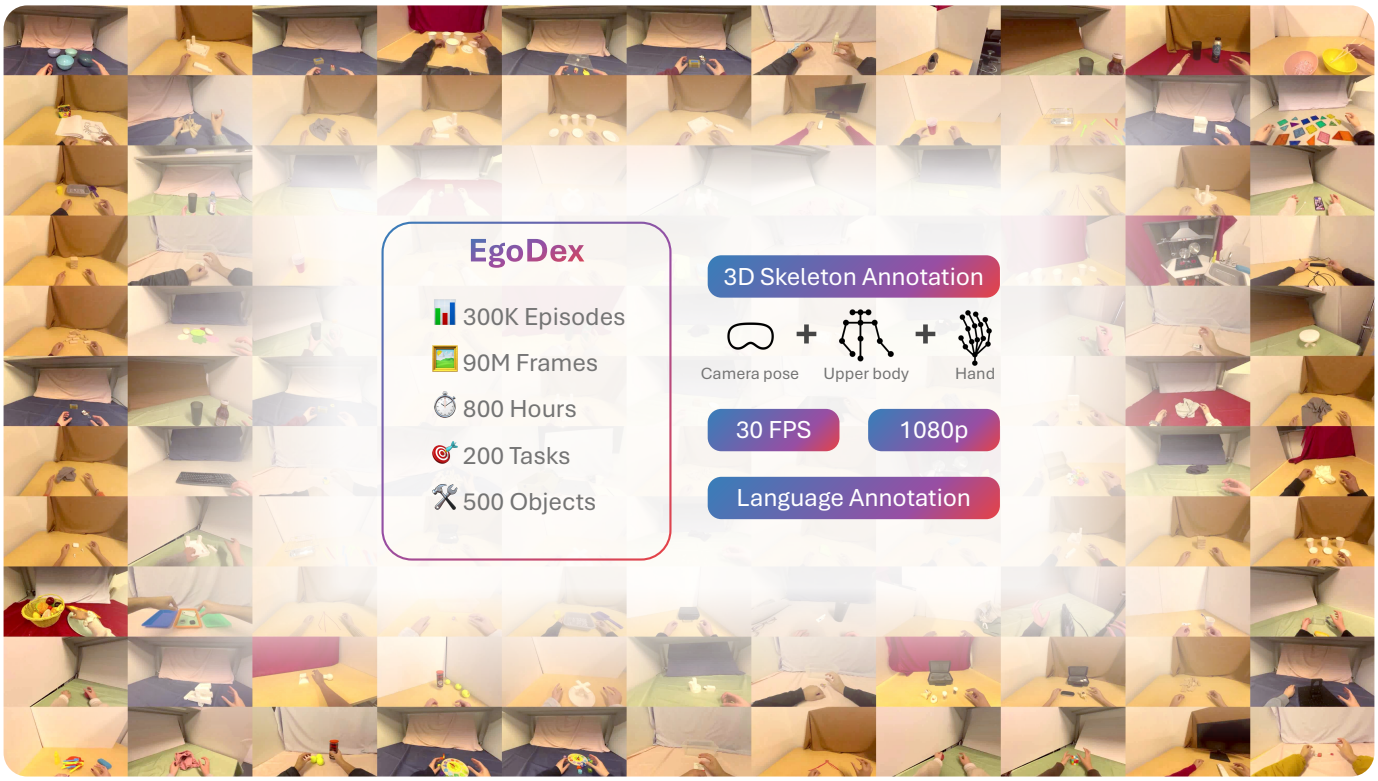

Fig. 1: **EgoDex** is a large-scale egocentric dataset focusing on human dexterous manipulation.

***Abstract*—Imitation learning for manipulation has a well-known data scarcity problem. Unlike natural language and 2D computer vision, there is no Internet-scale corpus of data for dexterous manipulation. One appealing option is egocentric human video, a passively scalable data source. However, existing large-scale datasets such as Ego4D do not have native hand pose annotations and do not focus on object manipulation. To this end, we use Apple Vision Pro to collect EgoDex: the largest and most diverse dataset of dexterous human manipulation to date. EgoDex has 829 hours of egocentric video with paired 3D hand and finger tracking data collected at the time of recording, where multiple calibrated cameras and on-device SLAM can be used to precisely track the pose of every joint of each hand. The dataset covers a wide range of diverse manipulation behaviors with everyday household objects in 194 different tabletop tasks ranging from tying shoelaces to folding laundry. Furthermore, we train and systematically evaluate imitation learning policies for hand trajectory prediction on the dataset, introducing metrics and benchmarks for measuring progress in this increasingly important area. By releasing this large-scale dataset, we hope to push the frontier of robotics, computer vision, and foundation models.**

I. Introduction

The "bitter lesson" [36] of recent breakthroughs in large language models and large vision models is that the simple recipe of supervised learning with vast amounts of data is far more effective than competing approaches. Two key challenges have prevented the application of the bitter lesson to the longstanding challenge of autonomous robot manipulation: (1) it is unclear what data should be collected, and (2) it is unclear how such data can be collected at the requisite scale.

The leading approach to data collection for robot imitation learning is teleoperation, in which human operators provide demonstrations by directly controlling robot hardware. Recent works such as Open X-Embodiment [26] and DROID [16] pioneer community-wide efforts to pool together hundreds of hours of robot teleoperation data. While such datasets can be used effectively for pretraining robot control policies, teleoperation is bottlenecked by physical robot operation, and it is unclear how to continue scaling this paradigm beyond its current size. Other works explore learning visual representations from existing in-the-wild Internet videos and images [32, 20]. In this case, while large-scale data is available, unstructured video data lacks the precise annotation necessary to learn dexterous manipulation.

We explore a middle path between the two: egocentric human video with paired 3D hand pose annotations. As sug-

gested by recent work [15, 30], such an approach is *passively scalable*, similar to text and images on the Internet. Effectively learning from such data is critical in a future where wearable headsets and smart glasses may be omnipresent. Data is a crucial component for doing so; before AlexNet [17] must come ImageNet [34].

To this end, we introduce EgoDex: a large-scale dataset and benchmark for learning dexterous manipulation from large-scale egocentric video. EgoDex consists of 829 hours of 30 FPS video and paired skeletal data with a total of 90 million frames and 338000 task demonstrations across 194 tabletop manipulation tasks. To our knowledge, the EgoDex dataset is the largest and most diverse dataset of dexterous human manipulation to date.

There are several key properties of the proposed data that make it more suitable for dexterous manipulation than existing alternatives:

- EgoDex is passively scalable, unlike robot teleoperation and other approaches that require deliberate effort for data collection. EgoDex suggests the human hand as a common embodiment, unlike teleoperation and other approaches that collect data that is only compatible with specific robot hardware platforms.
- EgoDex has 30 FPS 1080p egocentric video with a wide field of view, capturing much of what a human sees while manipulating objects. It has precise and highly detailed 3D pose information for the user's head, arms, wrists, and each joint of each finger from on-device SLAM and calibrated cameras, containing critical dexterous manipulation data unlike in-the-wild Internet videos and Ego4D [11].
- EgoDex consists of extremely diverse behaviors beyond simple pick-and-place such as unscrewing a bottle cap, flipping pages of a book, and plugging a charger into a socket. It consists entirely of active manipulation, unlike existing large egocentric video datasets such as Ego4D.

We systematically evaluate imitation learning policies for hand trajectory prediction to assess the state of the art and identify challenges for future research. We hope that EgoDex will not only accelerate progress in robot manipulation but also be useful more broadly in applications such as augmented reality, computer vision, assistive prosthetics, and human-computer interaction.

II. Related Work

A. Large-Scale Manipulation Datasets

Several prior works introduce large-scale open-source robot teleoperation datasets including RoboTurk [21], BridgeData [38], RT-X [26], and DROID [16]. While such datasets contain up to hundreds of hours of valuable manipulation data, it is not clear how to scale the paradigm further than its present scale. Robot teleoperation is extremely labor-intensive and resource-constrained, requiring an operational physical robot and a human teleoperator actively controlling the robot to perform each desired task. Furthermore, it is not clear to what degree such datasets can generalize beyond the set of hardware embodiments and camera viewpoints with which they were collected, even when the datasets consist of samples collected across multiple different embodiments.

Other large-scale datasets such as Ego4D [11] and EPIC-KITCHENS [8] consist of egocentric video recording humans perform various activities. While such datasets are more scalable and not limited to particular hardware platforms, they typically do not focus on manipulation and do not have paired 3D annotations for dexterous manipulation.

There is also a large body of work that considers hand-object interaction [19, 2, 4]. While these datasets often do have 3D hand pose annotations, they are orders of magnitude smaller than EgoDex due to manual annotation processes. Moreover, their emphasis is primarily on grasping rather than diverse and long-horizon manipulation tasks.

B. Scalable Methods for Robot Data Collection

Recent work identifies the data scarcity problem in robot imitation learning and proposes innovative techniques for scalable data collection. Chi et al. [7] propose the "universal manipulation interface": handheld grippers that enable human teachers to provide demonstrations without physical robots. Wang et al. [39] introduce a portable data collection system with motion capture gloves. Others propose collecting robot-free demonstrations by simulating robot hardware in augmented reality [6, 27, 23].

These approaches all face a similar pitfall: they require *active* data collection. While they may make it easier to collect data than teleoperation, human demonstrators must still be incentivized to intentionally collect the data. Such approaches face a significant uphill battle in approaching the scale of Internet datasets, where text and images are not deliberately collected but rather a passive byproduct of human interaction with the Internet.

C. Learning from Human Video

Video data is abundant on the Internet. Prior work explores representation learning on unstructured large-scale image and video data for pretraining visual encoders [32, 20] and grasp affordances [1] for downstream manipulation. However, raw unstructured video data faces a prohibitively large gap between its image distribution and that of a dexterous manipulation task. Moreover, such videos are not labeled with corresponding motor actions with which to train a policy.

One option is to postprocess the unstructured video data with 3D hand prediction networks such as HaMeR [28], recently explored by Ren et al. [33]. However, the prediction quality of these networks can suffer without multiple viewpoints and detailed knowledge of the camera extrinsics at all times, usually unavailable with raw Internet video. In contrast, the EgoDex dataset includes 3D head and hand tracking *at the time of collection*, where multiple cameras on the Vision Pro, known intrinsics and extrinsics, and a production-grade hand prediction network all contribute to precise annotation.

Most similar to our work is EgoMimic [15], which proposes the collection of egocentric video and paired 3D hand tracking.

Dataset	# Traj.	# Tasks	# Frames	Lang. Annot.	Cam. Calib.	Dexterous Annot.	Collection Method
RoboTurk [21]	2k	3	12M	✗	✗	✗	teleoperation
RoboNet [9]	162k	n/a	15M	✗	✗	✗	scripted
BridgeData V2 [38]	60k	13	2M	✓	✗	✗	teleop+scripted
DROID [16]	76k	86	19M	✓	✓	✗	teleoperation
EgoMimic [15]	2k	3	0.4M	✗	✓	✗	egocentric video
EPIC-KITCHENS [8]	40k	125	12M	✓	✗	✗	egocentric video
HOI4D [19]	4k	54	2M	✗	✗	✓	egocentric video
Ego4D (HOI) [11]	89k	n/a	21M	✓	✗	✗	egocentric video
EgoDex (ours)	**338k**	**194**	**90M**	✓	✓	✓	egocentric video

TABLE I: Comparison of different robot manipulation datasets (above middle line) and human manipulation datasets (below middle line). Ego4D (HOI) considers the subset of Ego4D that involves hand-object interaction. EgoDex has the largest amount of trajectories, tasks, and frames by a large margin and has language annotation, camera calibration, and dexterous annotation. "Dexterous annotation" is defined here as labels for multi-finger hand poses, which does not include lower fidelity pose data like parallel jaw robot grippers or wrist-only tracking.

The primary difference is scale: while EgoMimic collects around 4 hours of data, we collect 829 hours with a much broader data and task distribution. We also collect more dexterous annotations, critical for downstream manipulation: 3D positions and orientations for the upper body including the head, shoulders, arms, and 25 joints in each hand, whereas EgoMimic collects only the wrist positions. Lastly, a preview of EgoDex appeared in Qiu et al. [30], which used a 3% subset of EgoDex for successful human-to-robot transfer.

III. EgoDex Dataset

A. Overview

The EgoDex dataset contains 829 hours of 1080p, 30 Hz egocentric video with 338000 episodes across 194 tasks. This is a total of 90 million frames (i.e., data samples). The full dataset takes 2.0 TB of storage on disk. We compare EgoDex to existing manipulation datasets in Table I. EgoDex has almost an order of magnitude more data than the next largest dataset as well as higher task diversity. It also has language annotations, camera extrinsics, and dexterous annotations (Section III-C).

B. Data Collection

All data is collected with Apple Vision Pro running visionOS 2. The high-resolution and high-frequency passthrough and wide field of view enable intuitive egocentric data collection, where the collector can observe the environment unobstructed as if with their own eyes, and the camera data records precisely what the collector sees without any pose offsets (unlike, for instance, a head-mounted camera). Production-grade pose tracking software enables natural demonstration with bare hands and without any additional hardware apparatus. Video data is saved with AVFoundation and pose data is tracked by ARKit. The data was collected by 10 operators in tabletop environments.

To streamline data collection, data is recorded in *sessions*: approximately 10-15 minute segments that consist of many individual episodes, where episode boundaries are indicated by a "pause" and subsequent "resume" of recording from the data collection app. Raw video is compressed to facilitate data transfer, upload, download, and storage. Without the use of modern video compression algorithms, the raw data would take over 500 TB of disk space, about 250× its current size. At training time, data is loaded efficiently with PyTorch torchcodec [37], which only decodes the desired frames in the sampled batch of data.

C. Modalities

The data consists of the following: 1) Egocentric RGB video with 1920×1080 resolution at 30 Hz frequency. 2) Camera intrinsics and extrinsics at 30 Hz. 3) Position and orientation of all upper body joints (including 25 joints for each hand) at 30 Hz. 4) Confidence values for pose predictions at 30 Hz. 5) Natural language annotation of the manipulation.

The metadata annotated by data collectors includes the task name, a brief task description in natural language, details about the environment, and details about the object(s) that are manipulated. Since the metadata can be noisy, these fields are provided as input to GPT-4 [25], which combines this information into a single natural language description.

Confidence values are scalars between 0 and 1 indicating the ARKit prediction confidence per skeletal joint. A confidence of zero indicates that the joint is fully occluded from view. See Appendix B for a comprehensive list of all the joints and more information.

D. Task Types

EgoDex consists of 3 types of tasks:

- *Reversible* tasks are pairs of tasks that are the inverse of each other. The distribution of final states for one task is within the distribution of initial states for its inverse. For

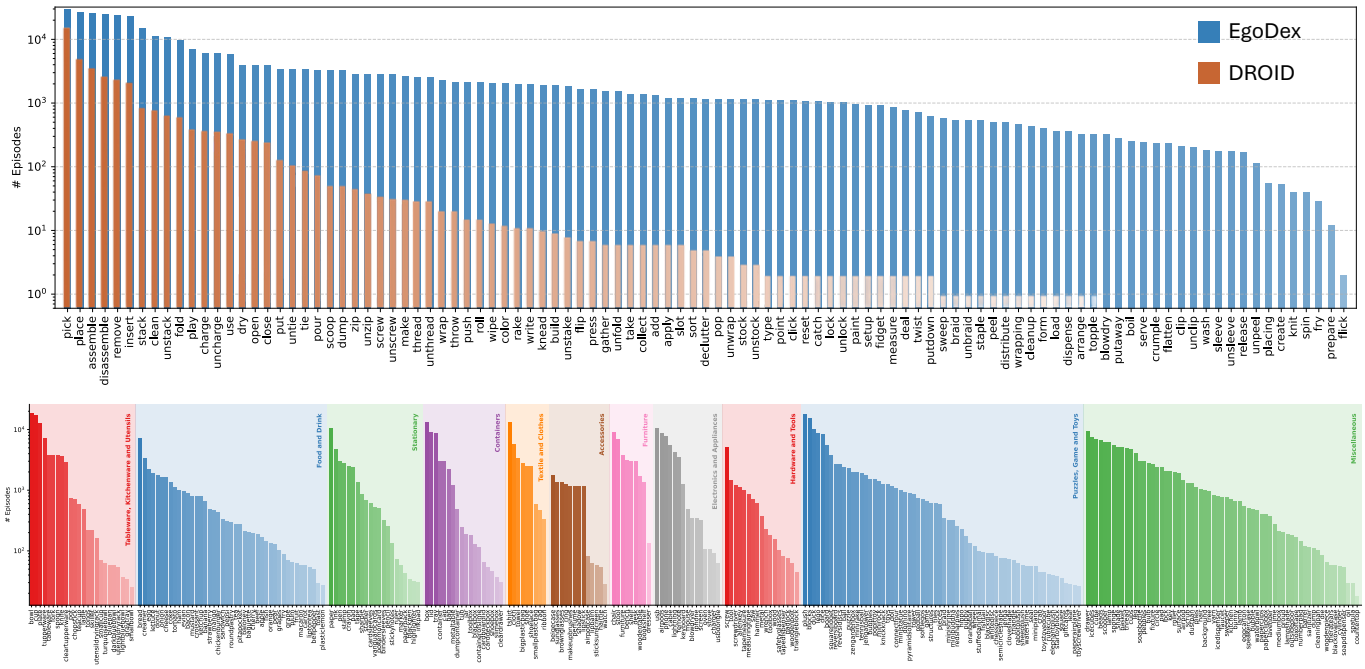

Fig. 3: Distribution of EgoDex dataset. **Top**: Distribution of distinct verbs, sorted by frequency. The horizontal axis are verbs of EgoDex. The orange plot is taken from DROID [16]. While many verbs in DROID are below the 10^1 mark, most verbs in EgoDex are above the 10^3 mark. **Bottom**: Distribution of distinct objects. The clustering is suggested by GPT-4.

example, connecting a charger to a device and removing a charger from the device.

- *Reset-free* tasks are tasks with a final state distribution that falls within its *own* initial state distribution. For example, throwing a ball in the air and catching it (where gravity acts as the reset).
- *Reset* tasks are tasks in which the environment must be reset to the initial state distribution after each demonstration.

Reversible and reset-free tasks enable a higher yield from data collection as they eliminate costly resets, which are not included in the recorded data.

E. Diversity

Prior work [16, 26] identify several potential axes of demonstration diversity: viewpoint diversity, task diversity, scene diversity, object diversity, and more. In EgoDex, the emphasis is diversity in *dexterous manipulation behaviors*. Tasks and objects vary such that the required dexterity ranges far beyond pick-and-place, the primary behavior in most robot teleoperation datasets. For example, tasks include tightening a screw, tying shoelaces, dealing cards, flipping pages, catching tennis balls, and slotting batteries. The task distribution covers a wide range of everyday household manipulation tasks that can be performed on a tabletop surface. There is also a significant amount of basic pick-and-place with diverse objects as well as the benchmark tasks from the FurnitureBench assembly benchmark [13]. The full list of 194 tasks provided in Appendix A.

To get a sense of the spread of the task distribution, as in prior work we plot the distribution of de-duplicated verbs in Figure 3. We observe that the distribution is much wider than prior works such as DROID [16], where a large fraction of verbs have less than 10^1 demonstrations and sometimes only a single demonstration; in contrast, most of the verbs in EgoDex have more than 10^3 demonstrations.

Still, the verb distribution does not capture the full diversity of manipulation behaviors or tasks. For example, "assemble" can involve radically different behaviors in the context of different objects and tasks. See Figure 4 for examples of different dexterous manipulation behaviors captured in the dataset.

While the *scene* diversity in EgoDex is limited to tabletop environments, the Cartesian product of scene and behavior is not the focus of our work, which focuses on behavioral diversity. Scene diversity can be introduced with modern visual data augmentation methods such as image-to-image generative models [42, 5].

F. Access

The full dataset has been made publicly available for download, but the hosting site breaks anonymity for the double-blind reviews. Data will be provided after reviews conclude. The dataset is licensed under CC-by-NC-ND terms.

IV. EgoDex Benchmarks

A. Action Representation

Given the full set of skeletal joints in the EgoDex dataset, many action representations are possible: wrist positions, wrist orientation, positions of fingertips, and so on. Since we focus

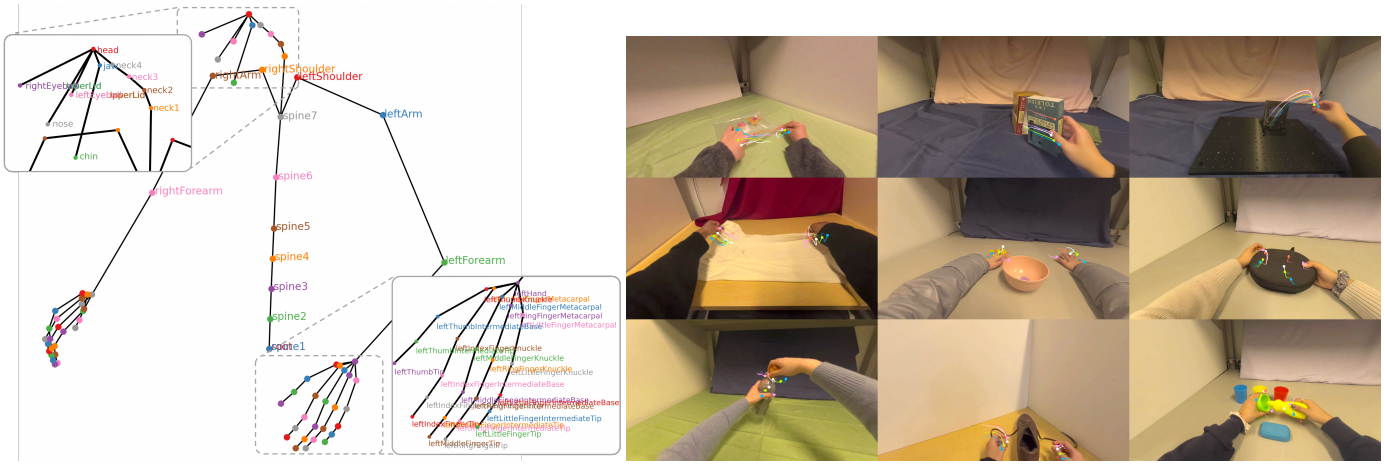

Fig. 4: **Left**: Joints captured by EgoDex. **Right**: Examples of dexterous manipulation behaviors. Tracked fingertips are highlighted in distinct colors and show 0.5 seconds of motion before the current frame. From left to right, top to bottom, the tasks are: unzipping a ziploc bag, removing a book from a bookshelf, removing a screw from a fixture, folding a t-shirt, decluttering, opening a case, unscrewing a bottle cap, tying shoelaces, and washing a cup.

on dexterous manipulation, we choose a representation that both captures sufficient bimanual dexterity. Specifically, the action $\mathbf{a_t}$ at time t is represented as the 3D position of each wrist, the 6D orientation of each wrist (where the 6 values are the first two columns of the rotation matrix), and the 3D position of each fingertip. Thus, each action has a total dimensionality of 2 hands $\times$ (3 + 6 + (3 $\times$ 5 fingertips)) = 48. In practice, actions are predicted in chunks over a fixed time horizon. All poses are represented in the camera frame.

B. Benchmarks

We propose two benchmark tasks for EgoDex. The first is *trajectory prediction*: from the egocentric image observations, skeletal joint poses, and natural language description, the task is to predict the trajectories of the hands for a given time horizon following the observations. Specifically, we seek to train the following estimator:

$$f_\theta(\mathbf{o}_{0..t}, \mathbf{s}_{0..t}, l) = \hat{\mathbf{a}}_{t:t+H}$$

where $\mathbf{o}_{0..t}$ are the egocentric image observations up to and including time t, $\mathbf{s}_{0..t}$ are skeletal pose observations up to and including time t, l is a natural language description, $\hat{\mathbf{a}}_{t:t+H}$ is the predicted action chunk, and H is the prediction horizon.

Since multimodality can be very severe in natural human motion, the second benchmark is *inverse dynamics*: from the image observations and skeletal poses up to time t as well as a goal image observation at the end of the time horizon, the task is to predict the trajectories of the hands in between the start and end observations. In this case, we train the following estimator, which can be interpreted as a visually goal-conditioned policy:

$$f_\theta(\mathbf{o}_{0..t}, \mathbf{s}_{0..t}, \mathbf{o}_{t+H}, l) = \hat{\mathbf{a}}_{t:t+H}$$

Each of these benchmarks are parameterized by prediction horizon H. For example, a short-horizon trajectory prediction task may set $H = 30$ (1 second), while a more difficult long-horizon task may set $H = 90$ (3 seconds).

Unlike typical robot hardware experiments that can vary across physical environments, the EgoDex benchmarks are fully reproducible with a fixed training and test set. We set aside 1% of the EgoDex dataset as a fixed held-out test set for evaluations, where the remaining 99% can be split across training and validation as desired.

C. Evaluation Metrics

Since trajectory prediction for natural human motion is inherently multimodal, evaluating a single predicted trajectory against the ground truth sample may be insufficient for measuring correctness. For example, for the simple task of placing a fruit in a basket, it could be placed at variable locations within the basket, moved at variable speeds, and moved in different but equally valid arcs from the initial position to the basket.

Thus, for each benchmark task we evaluate performance with a "best of K" metric. For each data point in the full test set, we sample the trained model K times to capture different possible modes. We then compute the distance between the ground truth trajectory and the trajectory closest to it out of the K samples, where "distance" is calculated as the Euclidean distance between predicted 3D keypoint positions and their ground truth 3D counterparts, averaged over each timestep in the predicted chunk and each of the 12 keypoints (i.e., the wrist and fingertips of each hand). Intuitively, this value can be interpreted as the average positional error in 3D space between ground truth and prediction in meters. The final value is averaged over the full test set. For deterministic models, the value is the same regardless of K; for stochastic models, the value improves as K increases, as the model gets more chances to sample the ground truth mode.

Model	Avg Distance (m)			Final Distance (m)		
	$K=1$	$K=5$	$K=10$	$K=1$	$K=5$	$K=10$
Dec + BC	0.045	0.045	0.045	0.062	0.062	0.062
Dec + DDPM	0.053	0.044	0.041	0.071	0.050	0.044
Dec + FM	0.052	0.042	0.040	0.071	0.049	0.043
EncDec + BC	**0.044**	0.044	0.044	**0.060**	0.060	0.060
EncDec + DDPM	0.052	0.042	0.039	0.071	0.048	0.043
EncDec + FM	0.051	**0.041**	**0.038**	0.070	**0.047**	**0.041**

TABLE II: Evaluations for different models on trajectory prediction with a 2 second horizon.

Model	Avg Distance (m)			Final Distance (m)		
	$H=30$ (1s)	$H=60$ (2s)	$H=90$ (3s)	$H=30$ (1s)	$H=60$ (2s)	$H=90$ (3s)
Dec + BC	**0.031**	0.045	0.053	**0.049**	0.062	0.069

TABLE III: Results for models trained and evaluated with different prediction horizons. As expected, accuracy falls as the prediction horizon increases. $H=60$ values are repeated from Table II for convenience.

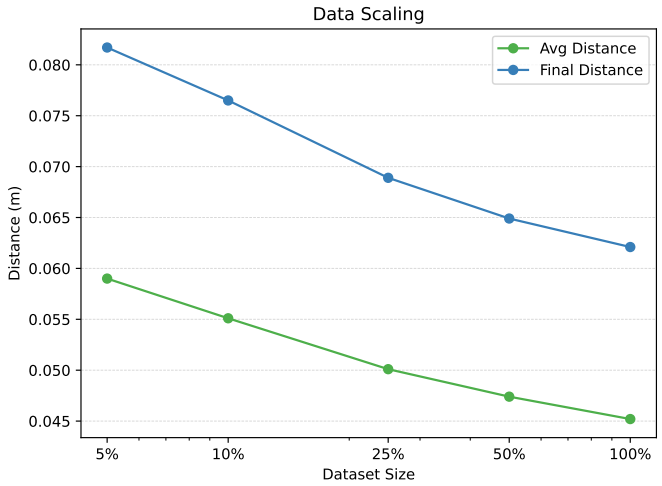

Fig. 5: Distance metrics w.r.t. training dataset size, where size is plotted on a log-scale. Performance improves as the dataset gets larger.

V. Experiments

We train and evaluate state-of-the-art imitation learning policies from the X-IL framework [14] on the benchmarks from Section IV. Specifically, we train two Transformer model architectures (encoder-decoder and decoder-only) and three policy representations (behavior cloning, denoising diffusion, and flow matching). We also run experiments to evaluate the effect of prediction horizon, visual goal-conditioning, dataset size, and model size. In total we train and evaluate 14 different models. We train all models for 50,000 gradient steps with a batch size of 2048 parallelized across 8 NVIDIA A100 GPUs. Additional training and model details are provided in Appendix C. The results are presented in Tables II, III, IV and Figure 5 and summarized below.

Model	Avg Distance (m)	Final Distance (m)
Dec + BC	0.045	0.062
Dec + BC w/ goal image	**0.035**	**0.029**

TABLE IV: Visual goal-conditioning results. Training a model with visual goal conditioning reduces average distance by 22% and final distance by 53%.

Encoder-decoder architectures outperform decoder-only. In Table II we observe that all encoder-decoder ("EncDec") models consistently outperform their decoder-only ("Dec") counterparts by a small margin.

Different policy representations excel in different settings. In Table II we observe that the encoder-decoder flow matching ("FM") model outperforms the other models for $K=5$ and $K=10$ by up to 34%. As expected, denoising diffusion ("DDPM") and FM evaluations improve as K increases, while behavior cloning ("BC") remains the same independent of K as it is deterministic. Note however that for the $K=1$ setting, BC outperforms both diffusion and flow-matching by about 15%. This suggests that the average prediction of BC is better than DDPM and FM, while the best prediction of DDPM and FM is better than BC's single prediction.

Performance degrades as the prediction horizon increases. In the remaining experiments we vary different properties while fixing the model to the simplest policy: decoder-only behavior cloning. In Table III we see that reducing the horizon from 2 seconds to 1 second improves average and final distance by 31% and 21% respectively, while increasing the horizon from 2 to 3 seconds worsens average and final distance by 18% and 11% respectively. Intuitively, accurate prediction becomes more challenging as the horizon increases as the model must predict 48-dimensional dexterous actions farther into the future.

Visual goal-conditioning significantly improves performance. In Table IV we observe that visual goal-conditioning reduces average distance by 22% and final distance by 53%. Intuitively, a visual goal provides a visual "anchor" to ground the endpoint of the predicted trajectory and mitigate multi-modality. This yields a baseline score for the inverse dynamics benchmark specified in Section IV.

Medium-size model capacity is sufficient for the current

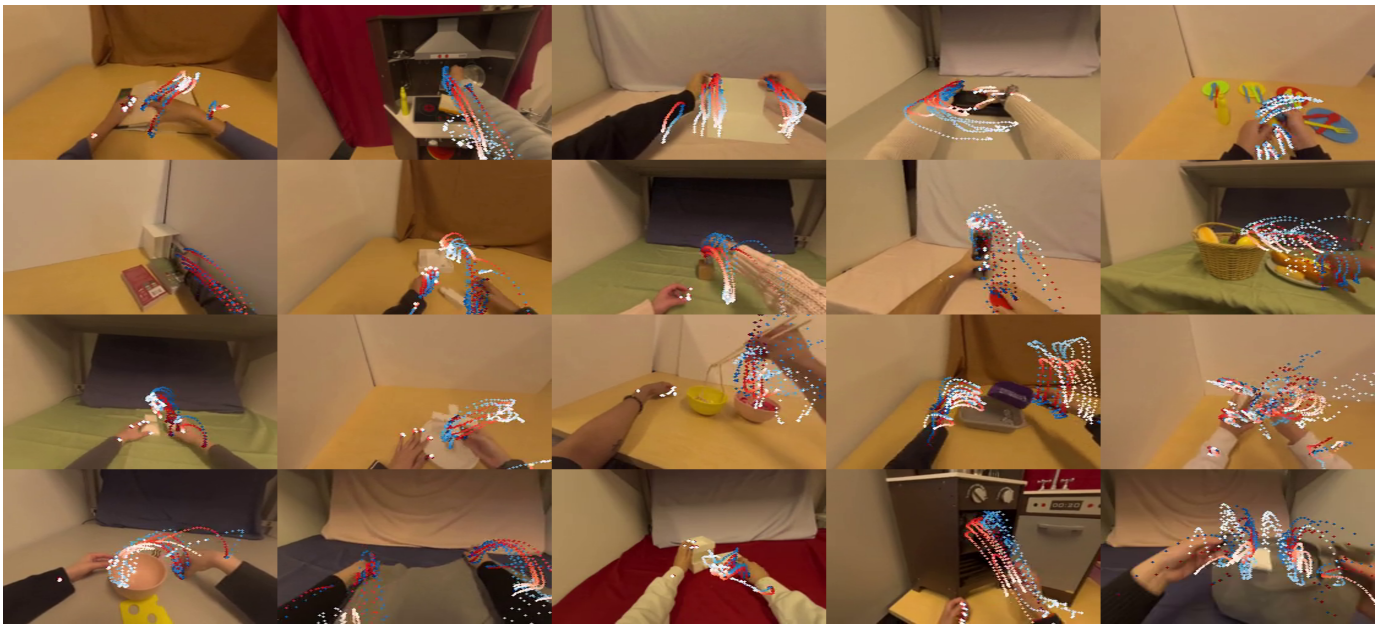

Fig. 6: Model prediction visualizations for Dec + BC on test set images with a 2 second horizon. Blue trajectories are ground truth and red trajectories are predictions, where darker colors and closer to the current frame and lighter colors are further in the future. The points shown are the wrist and fingertip positions projected into the camera frame.

dataset size. We train and evaluate a larger Dec+BC model with 500 million parameters as opposed to the default 200 million parameters. The larger model attains average distance 0.045 and final distance 0.062, exactly the same as the default 200 million parameter model. This may increase accessibility for the EgoDex benchmarks, as medium-size models fit comfortably on commodity GPU hardware.

Performance scales with dataset size. In Figure 5 we observe that average and final distance improve as dataset size increases. Results suggest that performance scales with data, motivating the collection of large-scale egocentric datasets like EgoDex.

VI. Research Use Cases

a) Robotics: Given a robot embodiment that mimics the visual appearance, kinematics, and dynamics of human arms and hands, a control policy mapping egocentric video to 3D arm and hand pose deltas such as those trained in Section V would be deployable *zero-shot* without any additional data or fine-tuning. Such an embodiment would lie within the training data distribution, both visually and dynamically.

While significant progress has been made in the development of robot hardware with humanoid morphologies and dexterous hands, there remains a prohibitive embodiment gap between humans and today's robots. Some options for bridging the embodiment gap include 1) co-training with a small-scale robot dataset, as demonstrated by [15, 30]; 2) pretraining with large-scale human data and supervised fine-tuning with smaller-scale robot data, similar to the training recipe for large language models; 3) training a visual encoder on the human data for more data-efficient imitation learning downstream, similar to R3M [22]; 4) learning robot manipulation priors from the human-object interaction trajectories and then fine-tuning with reinforcement learning or imitation learning [35, 10].

b) Perception: EgoDex can be used for learning tasks such as action recognition and human-object interaction detection from egocentric videos. Datasets like EPIC-KITCHENS [8] has demonstrated the value of egocentric video for recognizing and anticipating daily actions, and challenges have expanded to tasks like detecting active objects and predicting state changes from egocentric video. Researchers can also study which objects are involved in each action and how. For example, modeling the contact points, grasps, and trajectories when using a tool (screwdriver, scissors, etc.). A related task is learning object affordances, i.e., understanding what actions each object supports.

c) Video Generation and World Models: Recent advances in large-scale diffusion models have significantly enhanced the capabilities of language-conditioned video generation, producing temporally coherent and semantically precise visual narratives from natural language inputs [18, 29, 24]. These generative frameworks have demonstrated potential not only in creating realistic and detailed video content but also as world models for decision-making tasks, supporting reinforcement learning agents by simulating future outcomes based on predicted visual dynamics [40, 3, 41, 12]. Despite these impressive advancements, there remains a substantial research gap in video generation and world modeling from an egocentric viewpoint. Egocentric perspectives present unique challenges, including managing significant viewpoint variability, maintaining temporal and spatial coherence amid frequent

camera movements, and accurately reflecting agent-centric interactions and intentions. Since EgoDex provides annotations for 3D poses and language, it enables the possibility of training an egocentric foundation world model.

VII. Conclusion

We introduce EgoDex, a massive dataset of egocentric video paired with 3D pose annotations in a wide range of dexterous manipulation tasks. We train and evaluate imitation learning policies for hand trajectory prediction on this data.

While EgoDex has significant diversity across tasks and manipulation behaviors, it is limited in background and scene diversity. The dexterous annotations can also be imperfect, especially during heavy occlusion (e.g., towel folding) or very high speed motions, as they are themselves model predictions. Future work involves procedural background randomization on the existing data [42] as well as data collection in more diverse environments.

References

[1] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13778–13790, June 2023.

[2] Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, Jakob Julian Engel, and Tomas Hodan. HOT3D: Hand and object tracking in 3D from egocentric multi-view videos. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[3] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[4] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S. Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, Jan Kautz, and Dieter Fox. DexYCB: A benchmark for capturing hand grasping of objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[5] Lawrence Yunliang Chen, Chenfeng Xu, Karthik Dharmarajan, Muhammad Zubair Irshad, Richard Cheng, Kurt Keutzer, Masayoshi Tomizuka, Quan Vuong, and Ken Goldberg. Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning. *Conference on Robot Learning (CoRL)*, 2024.

[6] Sirui Chen, Chen Wang, Kaden Nguyen, Li Fei-Fei, and C Karen Liu. Arcap: Collecting high-quality human demonstrations for robot learning with augmented reality feedback. *arXiv preprint arXiv:2410.08464*, 2024.

[7] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Robotics: Science and Systems (RSS)*, 2024.

[8] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018.

[9] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. *Conference on Robot Learning (CoRL)*, 2019.

[10] Alexey Gavryushin, Xi Wang, Robert J. S. Malate, Chenyu Yang, Xiangyi Jia, Shubh Goel, Davide Liconti, René Zurbrügg, Robert K. Katzschmann, and Marc Pollefeys. MAPLE: Encoding Dexterous Robotic Manipulation Priors Learned From Egocentric Videos. *arXiv preprint arXiv:2504.06084*, 2025.

[11] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Christian Fuegen, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the World in 3,000 Hours of Egocentric Video. In *IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022.

[12] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *International Conference on Learning Representations (ICLR)*, 2019.

[13] Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. Furniturebench: Reproducible real-

world benchmark for long-horizon complex manipulation. *Robotics: Science and Systems (RSS)*, 2023.

[14] Xiaogang Jia, Atalay Donat, Xi Huang, Xuan Zhao, Denis Blessing, Hongyi Zhou, Han A. Wang, Hanyi Zhang, Qian Wang, Rudolf Lioutikov, and Gerhard Neumann. X-il: Exploring the design space of imitation learning policies, 2025.

[15] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video. *arXiv preprint arXiv/2410.24221*, 2024.

[16] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset. *Robotics: Science and Systems (RSS)*, 2024.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[18] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *Advances in Neural Information Processing Systems*, 37:56424–56445, 2024.

[19] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21013–21022, June 2022.

[20] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations (ICLR)*, 2023.

[21] Ajay Mandlekar, Jonathan Booher, Max Spero, Albert Tung, Anchit Gupta, Yuke Zhu, Animesh Garg, Silvio Savarese, and Li Fei-Fei. Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[22] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

[23] Nataliya Nechyporenko, Ryan Hoque, Christopher Webb, Mouli Sivapurapu, and Jian Zhang. Armada: Augmented reality for robot manipulation and robot-free data acquisition. *arXiv preprint arXiv:2412.10631*, 2024.

[24] NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[25] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory

Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[26] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[27] Younghyo Park, Jagdeep Singh Bhatia, Lars Ankile, and Pulkit Agrawal. Dexhub and dart: Towards internet scale robot data collection, 2024. URL https://arxiv.org/abs/2411.02214.

[28] Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3D with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[29] Xiangyu Peng, Zangwei Zheng, Chenhui Shen, Tom Young, Xinying Guo, Binluo Wang, Hang Xu, Hongxin Liu, Mingyan Jiang, Wenjun Li, Yuhui Wang, Anbang Ye, Gang Ren, Qianran Ma, Wanying Liang, Xiang Lian, Xiwen Wu, Yuting Zhong, Zhuangyan Li, Chaoyu Gong, Guojun Lei, Leijun Cheng, Limin Zhang, Minghao Li, Ruijie Zhang, Silan Hu, Shijie Huang, Xiaokang Wang, Yuanheng Zhao, Yuqi Wang, Ziang Wei, and Yang You. Open-sora 2.0: Training a commercial-level video generation model in 200k, 2025.

[30] Ri-Zhao Qiu, Shiqi Yang, Xuxin Cheng, Chaitanya Chawla, Jialong Li, Tairan He, Ge Yan, David J. Yoon, Ryan Hoque, Lars Paulsen, Ge Yang, Jian Zhang, Sha Yi, Guanya Shi, and Xiaolong Wang. Humanoid policy as human policy, 2025. URL https://arxiv.org/abs/2503.13441.

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. URL https://api.semanticscholar.org/CorpusID:231591445.

[32] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In

Conference on Robot Learning, pages 416–426. PMLR, 2023.

[33] Juntao Ren, Priya Sundaresan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning, 2025. URL https://arxiv.org/abs/2501.06994.

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[35] Himanshu Gaurav Singh, Antonio Loquercio, Carmelo Sferrazza, Jane Wu, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Hand-object interaction pretraining from videos. *arXiv preprint arXiv:2409.08273*, 2024.

[36] Richard S. Sutton. The bitter lesson, 2019. URL http://www.incompleteideas.net/IncIdeas/BitterLesson.html. Accessed: 2025-02-26.

[37] PyTorch Team. torchcodec: Easy and efficient video decoding for pytorch. https://github.com/pytorch/torchcodec, 2024. Accessed: 2025-04-01.

[38] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

[39] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C. Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. In *Robotics: Science and Systems (RSS)*, 2024.

[40] Lirui Wang, Kevin Zhao, Chaoqi Liu, and Xinlei Chen. Learning real-world action-video dynamics with heterogeneous masked autoregression, 2025. URL https://arxiv.org/abs/2502.04296.

[41] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

[42] Chengbo Yuan, Suraj Joshi, Shaoting Zhu, Hang Su, Hang Zhao, and Yang Gao. Roboengine: Plug-and-play robot data augmentation with semantic robot segmentation and background generation, 2025. URL https://arxiv.org/abs/2503.18738.

APPENDIX

A. Complete List of Tasks

We provide a complete list of task names here, labeled as they appear in the dataset and separated by task type (reversible, reset-free, or reset, with definitions in Section III-D). Recall that each reversible task is actually a pair of two tasks. There are a total of 194 tasks. See Figure 7 for a visual of a subset of the objects used in the various manipulation tasks.

Reversible (76 $\times$ 2 total tasks):

- `braid_unbraid`
- `charge_uncharge_airpods`
- `deal_gather_cards`
- `fry_bread`
- `assemble_disassemble_furniture_bench_chair`
- `assemble_disassemble_furniture_bench_drawer`
- `assemble_disassemble_furniture_bench_square_table`
- `fold_unfold_paper_basic`
- `insert_remove_furniture_bench_cabinet`
- `gather_roll_dice`
- `insert_remove_airpods`
- `insert_remove_drawer`
- `insert_remove_shirt_in_tube`
- `insert_remove_usb`
- `load_dispense_ice`
- `open_close_insert_remove_tupperware`
- `pick_up_and_put_down_case_or_bag`
- `put_away_set_up_board_game`
- `screw_unscrew_fingers_fixture`
- `sleeve_unsleeve_cards`
- `stack_unstack_cups`
- `thread_unthread_bead_necklace`
- `tie_and_untie_shoelace`
- `insert_remove_tennis_ball`
- `open_close_insert_remove_case`
- `pick_place_food`
- `put_in_take_out_glasses`
- `screw_unscrew_allen_fixture`
- `set_up_clean_up_chessboard`
- `slot_batteries`
- `stack_unstack_bowls`
- `stack_unstack_tupperware`
- `throw_collect_objects`
- `vertical_pick_place`
- `wash_put_away_dishes`
- `add_remove_lid`
- `arrange_topple_dominoes`
- `assemble_disassemble_legos`
- `assemble_disassemble_soft_legos`
- `assemble_disassemble_structures`
- `assemble_disassemble_tiles`
- `boil_serve_egg`
- `build_unstack_lego`

- charge_uncharge_device
- clip_unclip_papers
- crumple_flatten_paper
- fry_egg
- assemble_disassemble_furniture_bench_desk
- assemble_disassemble_furniture_bench_lamp
- assemble_disassemble_furniture_bench_stool
- fold_stack_unstack_unfold_cloths
- fold_unfold_paper_origami
- insert_remove_furniture_bench_round_table
- insert_remove_bagging
- insert_remove_cups_from_rack
- insert_remove_plug_socket
- insert_remove_utensils
- lock_unlock_key
- open_close_insert_remove_box
- scoop_dump_ice
- screw_unscrew_bottle_cap
- setup_cleanup_table
- stock_unstock_fridge
- stack_unstack_plates
- throw_and_catch_ball
- tie_untie_rubberband
- wrap_unwrap_food
- zip_unzip_bag
- zip_unzip_case
- assemble_disassemble_jigsaw_puzzle
- stack_unstack_tetra_board
- stack_remove_jenga
- insert_dump_blocks
- rake_smooth_zen_garden
- play_reset_connect_four
- insert_remove_bookshelf

Reset-free (28 total tasks):

- color
- fidget_magnetic_spinner_rings
- measure_objects
- staple_paper
- use_rubiks_cube
- wash_kitchen_dishes
- wipe_screen
- knead_slime
- point_and_click_remote
- type_keyboard
- clean_surface
- dry_hands
- play_mancala
- flip_coin
- flip_pages
- paint_clean_brush
- play_piano

Fig. 7: Some of the objects used in the various manipulation tasks.

- push_pop_toy
- put_toothpaste_on_toothbrush
- wash_fruit
- wipe_kitchen_surfaces
- stamp_paper
- blowdry_hair
- knit_scarf
- makeup
- write
- clean_cups
- roll_ball

Reset (14 total tasks):

- clean_tableware
- declutter_desk
- basic_pick_place
- stack
- make_sandwich
- peel_place_sticker
- sweep_dustpan
- wrap
- assemble_jenga
- basic_fold
- pour
- sort_beads
- use_chopsticks
- play_reversi

B. Complete List of Skeletal Joints

The annotations consist of SE(3) poses (represented as 4 $\times$ 4 homogeneous transformation matrices) for each of the following joints, labeled by their names as they appear in the dataset:

Upper Body:

`hip, spine1, spine2, spine3, spine4, spine5, spine6, spine7, neck1, neck2, neck3, neck4, leftShoulder, leftArm, leftForearm, leftHand, rightShoulder, rightArm, rightForearm, rightHand`

Left Hand:

leftIndexFingerIntermediateBase, leftIndexFingerIntermediateTip, leftIndexFingerKnuckle, leftIndexFingerMetacarpal, leftIndexFingerTip, leftLittleFingerIntermediateBase, leftLittleFingerIntermediateTip, leftLittleFingerKnuckle, leftLittleFingerMetacarpal, leftLittleFingerTip, leftMiddleFingerIntermediateBase, leftMiddleFingerIntermediateTip, leftMiddleFingerKnuckle, leftMiddleFingerMetacarpal, leftMiddleFingerTip, leftRingFingerIntermediateBase, leftRingFingerIntermediateTip, leftRingFingerKnuckle, leftRingFingerMetacarpal, leftRingFingerTip, leftThumbIntermediateBase, leftThumbIntermediateTip, leftThumbKnuckle, leftThumbTip

Right Hand:

rightIndexFingerIntermediateBase, rightIndexFingerIntermediateTip, rightIndexFingerKnuckle, rightIndexFingerMetacarpal, rightIndexFingerTip, rightLittleFingerIntermediateBase, rightLittleFingerIntermediateTip, rightLittleFingerKnuckle, rightLittleFingerMetacarpal, rightLittleFingerTip, rightMiddleFingerIntermediateBase, rightMiddleFingerIntermediateTip, rightMiddleFingerKnuckle, rightMiddleFingerMetacarpal, rightMiddleFingerTip, rightRingFingerIntermediateBase, rightRingFingerIntermediateTip, rightRingFingerKnuckle, rightRingFingerMetacarpal, rightRingFingerTip, rightThumbIntermediateBase, rightThumbIntermediateTip, rightThumbKnuckle, rightThumbTip

Note that `leftHand` and `rightHand` refer to the wrists. Note also that the joint confidence values in the data behave differently for the wrists and the hands. Wrist confidence values (for `leftHand` and `rightHand`) indicate whether each hand is detected as a whole, while finger joint confidence values indicate confidence *relative* to the wrist. If, for instance, the left index fingertip has high confidence but the left wrist has low confidence, it is unlikely that the left index fingertip is reliable.

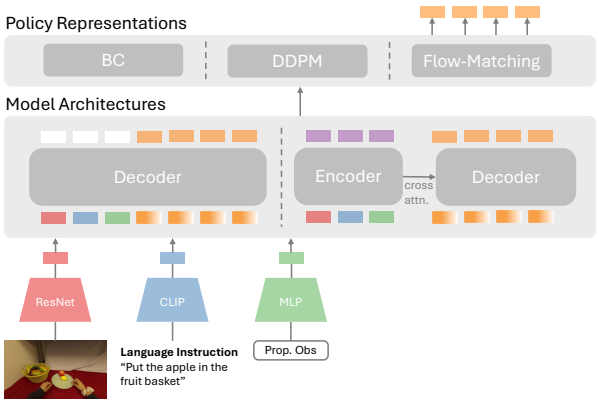

Fig. 8: Model architectures.

C. Training Details

In the experiments section we train and evaluate 14 different models: 6 combinations of architectures and policy optimization methods, 4 additional models with different training dataset sizes, 2 additional models with different prediction horizons, 1 model with a larger model size, and 1 model with visual goal-conditioning. See Figure 8 for intuition on the model architecture.

Each model is trained and evaluated on a single node with 96 logical CPUs (48 physical CPUs) and 8 NVIDIA A100 GPUs each with 80GB RAM. Training is run for 50,000 gradient steps with a batch size of 2048 (256 per GPU with data parallelism), at which point training and validation loss plateau. The full training run takes approximately 72 hours. The models are optimized with Adam and a learning rate of 1e-4. Decoder-only Image observations are resized to 224 $\times$ 224 and sent through a pretrained ResNet encoder, while language annotations are passed through a frozen CLIP [31] encoder. DDPM and FM models are trained and evaluated with 16 sampling steps. All other hyperparameters are the defaults from the X-IL codebase [14].