FIDO the Robot Dog: Learning Dynamics Models via In-the-Wild Fetch

Michelle Pan[1] **Maximilian Du**[1] **Sriram Yenamandra**[1] **Ajay Sridhar**[1]
Mark Zolotas[2] **Dorsa Sadigh**[1] **Jennifer Grannen**[1]

[1]Stanford University [2]Toyota Research Institute

mpan8@stanford.edu

Abstract: Learning generalizable dynamics models requires diverse, grounded training data. Existing approaches trained on internet-sourced data are constrained by the availability of high-quality samples, while the diversity of real-world data collection efforts is limited by infrastructure and operator demands. In our work, we propose collecting dynamics data through natural, embodied robot interactions. We present Fetch Interaction for Dynamics Observation (FIDO), a robot quadruped that collects in-the-wild dynamics data by playing fetch. FIDO first observes a ball trajectory thrown by a user. Then, it retrieves the ball and returns it, creating a naturally engaging and repeatable interaction that yields grounded observations of object dynamics, including the arc of throws, the effect of gravity, and the variability of real-world conditions. We collect FIDO data across N=18 users in three in-the-wild environments. A projectile dynamics model trained on this data exhibits strong transfer to unseen domain-adjacent tasks. FIDO outperforms a pretrained VLM baseline by 29.1% on shuttlecock trajectory prediction in badminton and by 26.8% on ball trajectory prediction in baseball, with both results improving monotonically with the diversity of collected fetch data. Our learned dynamics model is also directly useful for robot planning: a robot plans through these learned dynamics to catch a balloon before it hits the ground, where the addition of FIDO data improves catching performance by 20.0%. These results suggest that in-the-wild interaction diversity is a key driver of learning generalizable physical dynamics. Additional visuals can be found at our website https://fido-robot.github.io/site.

Keywords: Crowdsourcing Data, In-the-Wild Deployment, Dynamics Modeling

1 Introduction

Prior work has explored crowdsourcing as a path to data diversity. Human participation can yield large and varied datasets, but curating a useful dataset often requires significant infrastructure, operator expertise, or explicit incentive mechanisms [1, 2]. These demands make in-the-wild collection difficult in practice. One source of this overhead is that crowdsourced robotics data are often framed as soliciting expert task demonstrations [2]. Unlike imitation learning, however, dynamics learning does not require every interaction to demonstrate expert or optimal behavior [3]: a casual interaction still provides valid observations of how the world responds.

We propose using natural human–robot interactions as a data-collection mechanism. When an interaction is easy to understand, engaging, and naturally repetitive, participants generate dynamics-relevant observations as a byproduct of interacting with the robot. This produces data that is intentionally aligned with the target physical phenomenon while retaining the variation that arises across people and environments.

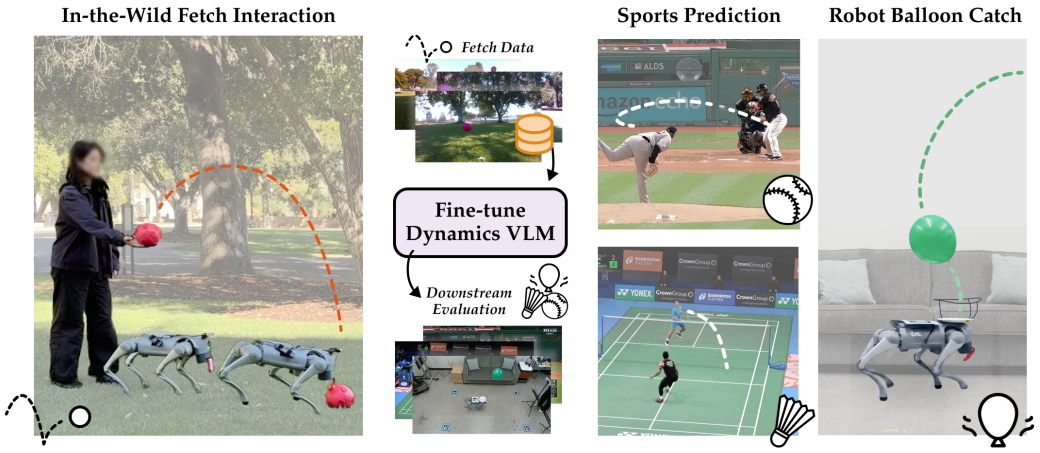


Figure 1: **FIDO: Fetch Interaction for Dynamics Observation.** We gather in-the-wild fetch data through crowdsourced interactions in a variety of environments [**Left**]. We fine-tune a Vision-Language Model (VLM) on the collected dynamics observations and demonstrate its generalizability [**Center**]. We evaluate our dynamics model on a real robotics task of catching a balloon mid-air, and two domain-adjacent sports tasks of badminton and baseball trajectory prediction [**Right**].

We instantiate this principle in FIDO (**F**etch **I**nteraction for **D**ynamics **O**bservation), a quadruped robot that plays fetch with a human partner. During each round, the robot observes a thrown ball, retrieves it, and returns it to the participant. Fetch provides a simple experimental setting for studying whether casual interaction can yield transferable dynamics knowledge. Each round produces observations of physically meaningful events: the arc of a throw, the pull of gravity, and the bounce on landing. The resulting data captures a greater variety of physical conditions than laboratory collection allows (e.g., backgrounds, lighting and weather conditions, surface types, and throwing styles). Unlike passive internet video, active collection from FIDO makes relevant physical events directly elicitable, rather than contingent on their availability in existing footage. Because collection can take place outside a controlled setting and does not depend on a fixed set of trained operators, we refer to it as *in-the-wild*. Our key contribution is an intuitive system for in-the-wild data collection that turns casual human play into observations of real-world object dynamics.

Using the FIDO system, we collect data across 18 interaction partners and three environments. We fine-tune a Vision Language Model (VLM) for dynamics prediction on this data (§4) and verify its generalizability in two experimental setups. First, we demonstrate transfer to two unseen domain-adjacent tasks: shuttlecock trajectory prediction in badminton and ball trajectory prediction in baseball. Compared to the VLM before training on FIDO data, the FIDO-trained model achieves a **29.1%** and **26.8%** improvement on shuttlecock and baseball prediction respectively, with performance improving monotonically with data diversity (§5.2). Second, we show that the FIDO-trained model is effective on a real-world dynamics prediction task: catching a balloon in a basket held by the robot before it hits the ground (§5.3). FIDO achieves **20.0%** better catching performance than baselines trained without FIDO data. Together, these results show that data elicited through a simple human-robot interaction can improve dynamics prediction across unseen objects and tasks and support downstream real-world robot behavior.

2 Related Work

Learning Dynamics Models. Past approaches to learning dynamics models have focused on task-specific settings, relevant to a single robot or environment [4, 5, 6, 7, 8, 9, 10]. While effective within their target domains, their performance often degrades outside the training distribution. Large-scale foundation models have enabled more general-purpose modeling [11, 12], with recent works demonstrating that Vision-Language Models (VLMs) can capture diverse object dynamics and can be leveraged for planning [13, 14, 15, 16]. However, these models are often trained on broad corpora that

were not constructed to repeatedly expose a target physical event, such as object flight and impact [13, 17, 18]. Our work complements these model-centric advances by studying the source of adaptation data. The FIDO system elicits repeated object-motion events through a fetch interaction. We evaluate whether finetuning a VLM on trajectory prediction on this data supports transfer across environments, viewpoints, and object-motion regimes. We also compare the FIDO-trained model with a model fine-tuned on curated internet videos of object motion.

Crowdsourcing Data. Crowdsourcing is an appealing option for collecting large and diverse datasets across many application domains [19, 20, 21, 22]. In robot learning, Mandlekar et al. [1] use a mobile phone teleoperation interface to collect manipulation data remotely at scale. Mirchandani et al. [2] investigate incentive strategies for crowdsourcing data collection using a bimanual robot teleoperation system deployed in a public space. Other works have developed new hardware interfaces — sensorized grippers, portable motion capture systems, or egocentric wearables — to enable demonstration collection in the real world without requiring access to a physical robot [23, 24, 25]. These approaches broaden demonstration collection, but participants must still deliberately produce useful demonstrations through specialized hardware, interfaces, or collection protocols. FIDO changes the contribution required of the participant: rather than producing a prescribed task demonstration, the participant plays a familiar game, and dynamics observations arise from the resulting physical interaction. Because the learning target is the object motion rather than the participant's proficiency, nonexpert variation in ball trajectory remains useful in training.

In-the-Wild Robot Data Collection. Deploying robots in homes, outdoor spaces, and other uncontrolled environments exposes them to physical conditions that are difficult to enumerate in the laboratory [26, 27, 28, 29, 30]. These deployments provide not only a test of robustness but also a source of data for continued model adaptation. UMI [25] demonstrates that collecting manipulation demonstrations in diverse, real-world environments rather than fixed lab settings enables zero-shot generalization to novel environments and objects. UMI increases accessibility by decoupling demonstration collection from the target robot. FIDO takes a complementary approach: it retains the robot but reduces participant burden by embedding data collection in a familiar interaction.

Most closely related to our work, Grannen et al. [31] use an in-the-wild robot platform to collect data that continually improve a VLM's Optical Character Recognition (OCR) capabilities, and similarly demonstrate transfer to domain-adjacent settings. However, their evaluation centers on a static perceptual recognition task. Our work shares Grannen et al.'s view of in-the-wild deployment as an opportunity for continued model improvement, but connects the learned model back to robot action: we target *dynamics prediction* and deploy the model in a real-world robot balloon-catching task.

3 FIDO: A System for In-the-Wild Data Collection

Our method, Fetch Interaction for Dynamics Observation (FIDO), considers the setting of "fetch", in which a user throws a ball for the robot to retrieve. The naturally repetitive and unconstrained nature of this task yields a diverse set of video observations of thrown ball trajectories, which we use downstream to learn generalizable projectile-style dynamics models.

Hardware Setup. We use a Unitree Go2 quadruped robot as our data collection platform. The Go2's portability and ability to traverse varied terrain enables usage in diverse environments, ideal for in-the-wild data collection. We additionally mount an Intel RealSense D435i depth camera facing horizontally forward on the robot's head.

Since the Go2 lacks native manipulation capabilities, we design a custom end-effector to enable ball retrieval. Our mechanism consists of a magnet enclosed in a felt "tongue", secured by a 3D-printed mount in place of the stock LiDAR unit at the base of the robot's head. We attach magnets of opposite polarity on the surface of the ball, a foam dodgeball, creating a passive retrieval mechanism that does not require additional actuation (Fig. 2 Right). The zoomorphic embodiment of the Go2



Figure 2: **In-the-Wild Data Collection.** The FIDO system is a robot quadruped that plays fetch using a magnetic ball and a felt-covered "tongue" (§3). During this interaction, FIDO collects video data on the ball trajectory. We use these videos to fine-tune a VLM for dynamics prediction. (§4).

naturally affords the fetch interaction, lowering the barrier to crowdsourcing data by grounding data collection in a familiar activity that requires little training or instruction for the interaction partner.

Fetch Interaction. The interaction begins with the robot standing at the interaction partner's feet and facing away from them (Fig. 2). The interaction partner is instructed to throw the ball into view of the RealSense camera mounted on the robot head (approx. 70° horizontal FOV). Once the robot detects the ball with an off-the-shelf object detection model, it begins approaching the ball. Upon reaching a close range, the robot tilts forward to lower the "tongue" end-effector and attempts to magnetically attach to the ball by moving towards the last known position of the ball. The robot then returns to a neutral stance to lift the ball off the ground and reverses to verify success of the retrieval. If the ball is still visible after backing up, the robot reattempts the approach. Once the robot has successfully obtained the ball, it navigates back towards its initial position, using an object detection model to identify the interaction partner. Finally, the robot assumes a sitting position in front of the person to hand off the ball.

Implementation Details. The fetch behavior is implemented using a finite state machine which responds to perception and proprioception feedback. We use YOLO11n [32] as our off-the-shelf object detector to run real-time ball and person detection onboard the robot. For each fetch interaction, we record video data from the RealSense camera and save the segment from which the ball first appears in the frame, until the robot begins to move to fetch the ball. The robot approaches the ball by running a PD controller to perform visual servoing. We control the dog's gait using a pretrained locomotion policy provided through the high-level interface in the Unitree SDK [33].

4 Learning a Dynamics Model with FIDO

We use videos obtained from FIDO's fetch interactions to adapt a dynamics model that predicts future object locations given past locations. In this section, we first formulate the dynamics prediction problem, then we describe how we obtain necessary supervision from FIDO, and finally explain how we adapt a pretrained VLM using this supervision.

Problem Setting. We model object dynamics as future prediction in image space. At time t, the dynamics model f_θ observes a past image I_{t-h+1}, and a short history of object locations $H_t = \{p_{t-h+1}, \ldots, p_t\}$, where each location is represented by its pixel position in the image $p_t = (x_t, y_t)$. The model is further provided with a query specifying which future timesteps to predict: $\Delta = \{\delta_1, \delta_2, \ldots, \delta_K\}$. The dynamics model is expected to predict the corresponding set of future object positions $Y_t = \{p_{t+\delta_1}, \ldots, p_{t+\delta_K}\}$. We write the dynamics model as: $\hat{Y}_t = f_\theta(I_{t-h+1}, H_t, \Delta)$. Here, I_{t-h+1} provides visual context at the beginning of the trajectory, while the history H_t provides recent motion of the object. The offset query allows the model to be transferred to varied use cases without changing model architecture. For example, catching a falling balloon requires long-horizon predictions while predicting whether a shuttlecock will land in bounds might require shorter-horizon predictions.



Figure 3: **FIDO Data and Domain Adjacent Test Settings.** We evaluate the generalizability of dynamics models trained on FIDO data on three different settings. The domain-adjacent domains of Balloon Catch, Badminton, and Baseball differ substantially from the FIDO data in object appearance, scale, and aerodynamic properties, as well as in camera viewpoint and environmental context.

Trajectory Supervision with FIDO. We learn a dynamics model using trajectory supervision extracted from FIDO videos, broadly focusing on modeling the "throwing" motion. Each fetch interaction provides a video sequence $V = \{I_1, \ldots, I_T\}$ demonstrating motion of a ball thrown by an interaction partner. We use state-of-the-art object detection and tracking methods [34, 35] to obtain the trajectory of the ball in image space, $\{p_1, p_2, \ldots p_T\}$, where p_t is the tracked pixel location of ball center. We sample an observation at time t such that a trajectory contains a valid history window and enough timesteps into the future to construct a prediction target. For each sampled observation, we randomly choose the number and stride of future predictions, while ensuring that all requested future timesteps are available in the trajectory. This generates multiple training examples by varying the observation and the future offset query from a single trajectory. In our implementation the history contains five observed points, which we found in practice to contain the necessary information for predicting future trajectories, such as the velocity and initial position of the ball.

Vision-Language Model Adaptation to Dynamics Prediction. Our goal with FIDO is to build a generalizable dynamics model, enabling transfer to related projectile trajectory prediction settings. Hence, we instantiate the dynamics model as a VLM which incorporates priors from large-scale Internet data. While state-of-the-art VLMs can produce qualitatively plausible outputs on physical reasoning tasks, their answers are often numerically inaccurate [36]. In this work, we improve a pretrained VLM's ability to predict physically realistic trajectories using FIDO data. To adapt the pretrained model efficiently while retaining its broad visual priors, we train it using low-rank adaptation (LoRA) [37], which keeps the pretrained backbone fixed and learns a small set of trainable low-rank updates to selected weight matrices.

5 Experimental Results

We evaluate our FIDO-trained dynamics model on three tasks, as shown in Fig. 3. First, we test the generalizability of the learned dynamics across two unseen domain-adjacent tasks: shuttlecock trajectory prediction in badminton and ball trajectory prediction in baseball. Second, we demonstrate its utility for robot planning on a real robotics task of catching a falling balloon before it hits the ground. Across both settings, we investigate whether the diversity of in-the-wild fetch data is the key driver of generalization.

Model	Num. Throwers	Num. Envs.	Badminton Shuttlecock RMSE (↓)	Badminton Shuttlecock DTW (↓)	Baseball Ball RMSE (↓)	Baseball Ball DTW (↓)
Pretrain VLM	–	–	0.55 ± 0.08	2.50 ± 0.37	0.41 ± 0.06	3.17 ± 0.48
Classical predictor	–	–	0.79 ± 0.02	4.35 ± 0.16	0.59 ± 0.01	5.19 ± 0.15
Coordinates-only LLM	–	–	0.54 ± 0.05	2.83 ± 0.27	$\mathbf{0.30 \pm 0.03}$	$\mathbf{2.42 \pm 0.28}$
Internet-tuned VLM	–	–	0.52 ± 0.07	2.41 ± 0.37	0.38 ± 0.05	3.01 ± 0.40
FIDO-tuned VLM	2	1	0.59 ± 0.04	3.04 ± 0.26	0.63 ± 0.02	5.59 ± 0.17
	2	3	0.50 ± 0.05	2.53 ± 0.28	0.51 ± 0.03	4.39 ± 0.31
	18	3	$\mathbf{0.39 \pm 0.05}$	$\mathbf{1.87 \pm 0.26}$	$\mathbf{0.30 \pm 0.03}$	$\mathbf{2.35 \pm 0.25}$

Table 1. **FIDO's in-the-wild data collection yields more diverse data, leading to better domain-adjacent transfer.** We evaluate the generalizability of dynamics models on two unseen trajectory prediction domains: badminton shuttlecock [38] and baseball [39]. We report the average and 95% confidence intervals of root mean squared error (RMSE) and dynamic time warping (DTW). Fine-tuning on the most diverse set of fetch data across throwers and environments achieves the best performance across both metrics and both domains.

5.1 FIDO Dataset

The FIDO dataset consists of 75 videos collected across three environments with 18 interaction partners, solicited as passersby in public outdoor areas. Each video captures a single throw of the ball, and the dataset is gathered over 59 minutes of human interaction time. Additional dataset information and post-processing details are included in Appendix A.

5.2 Generalizable Dynamics Modeling

We consider two domain-adjacent trajectory prediction tasks to evaluate the generalizability of our fetch-data trained dynamics model as visualized in Fig. 3. The first is shuttlecock trajectory prediction in badminton, evaluated on the TrackNet dataset [38]. The second is ball trajectory prediction in baseball, evaluated on the MLB-YouTube dataset [39]. Both tasks share underlying physical structure with fetch, specifically projectile motion under gravity, but differ in object properties, camera perspectives, and environmental context, making them a meaningful test of out-of-domain generalization.

For badminton, we evaluate on 100 TrackNetV2 trajectory segments constructed from official shuttlecock annotations. For baseball, we evaluate on 77 pitch trajectories from MLB-Youtube, for which we extract frame-level ball locations ourselves. We fine-tune Qwen3-VL-2B-Instruct [11] on fetch data collected by FIDO using LoRA and evaluate the resulting dynamics model on both tasks. Data pre-processing details and optimization hyperparameters are provided in Appendix B.

Baselines. We compare a FIDO-trained model against four baselines: 1) the pretrained VLM; 2) a classical baseline, in which we tune a family of classical trajectory predictors using FIDO data and select the lowest RMSE variant; 3) a coordinates-only LLM, in which we fine-tune the model on FIDO trajectories while removing the image input to test whether the image adds useful information beyond the observed trajectory; 4) an internet-tuned VLM, fine-tuned on projectile trajectories extracted from YouTube videos found in the UCF101 [40] dataset. We provide full baseline implementation details in Appendix B.

Diversity Ablations. Finally, to investigate the role of data diversity in generalization, we conduct ablations by fine-tuning on equally-sized datasets of fetch data with varying diversity:

1. Fine-tuning with fetch data collected in **1 environment** with **2 interaction partners**,
2. Fine-tuning with fetch data collected across **3 environments** with the same **2 partners**,
3. Fine-tuning with FIDO data collected *in-the-wild* across **3 environments** and with **18 interaction partners**.

The datasets for (1) and (2) are disjoint from the original FIDO dataset and are collected with two authors serving as interaction partners. Additional analysis of the diversity in each dataset is provided in Section A. By comparing performance across these conditions, we aim to demonstrate that

diversity in data collection, both across interaction and environment, is a key driver of transfer to unseen domains.

Results. As shown in Table 1, we measure the success of our generalizability with two metrics: root mean squared error (RMSE) and dynamic time warping (DTW), both normalized to the ground-truth length of the predicted trajectory. We report normalized RMSE and normalized DTW averaged across evaluation trials and three independent training seeds. The reported intervals denote 95% hierarchical bootstrap confidence intervals over evaluation trajectories and training seeds. For fixed baselines (first two rows in Table 1) that are not trained in our experiments, intervals reflect evaluation-trajectory variability only. The VLM tuned on the full FIDO set achieves the best RMSE and DTW performance for badminton shuttlecock trajectory prediction, outperforming all baselines and lower-diversity fetch data ablations. We observe consistent improvement in both metrics with increasing data diversity, from a single environment and two throwers, to three environments, to the full in-the-wild collection. Overall, FIDO achieves a 29.1% improvement in RMSE and a 25.2% improvement in DTW over the pretrained baseline, and outperforms the best lower-diversity ablation by 22.0% in RMSE and 26.1% in DTW.

We observe a similar trend in the baseball setting, where model performance on both RMSE and DTW improves monotonically with the diversity of fetch data collected. However, unlike badminton, the pretrained baseline outperforms both lower-diversity fetch data ablations. We hypothesize that this is because baseball ball dynamics are far more common in internet-scale training data than badminton shuttlecock dynamics, giving the pretrained model a stronger prior for this domain. Fine-tuning on fetch data consisting of relatively homogeneous throw styles (from only two throwers) appears to overfit to fetch-specific trajectory patterns, degrading generalization to baseball dynamics. With sufficient diversity, however, this effect is overcome – a model fine-tuned with in-the-wild FIDO data outperforms the pretrained baseline by 26.8% and 25.9% across RMSE and DTW respectively.

5.3 Robot Balloon Catching

We consider the task of catching a thrown balloon in a basket attached to the head of the robot. This task tests both the generalization capabilities of FIDO to a novel dynamics situation and its applicability to a downstream robot task. Because the robot has a limited speed, it strongly benefits from proactively moving to where the balloon will land, which is predicted by our dynamics model. Thus, the catching success is a reflection of the performance of the dynamics model predictions.

We employ an angled, overhead Intel RealSense D435i camera with the full arena workspace in view, as demarcated by four Aruco tags. Similar to Section 3, the position of the balloon in the scene is tracked using YOLO11n [32]. A context of past balloon positions, as well as an initial image of the scene, are fed into the Qwen3-VL-2B-Instruct [11] dynamics model to predict the future trajectory of the balloon. The robot uses this trajectory to plan waypoints toward the predicted landing location (see Appendix C for details).

We evaluate performance across four VLM variants with increasing in-domain data exposure:

1. **Pretrain**, which measures out-of-the-box performance of the VLM without finetuning on balloon or FIDO data,
2. **FIDO**, which measures performance after finetuning with 75 videos of FIDO data,
3. **Few-Shot Balloon**, which measures performance after fine-tuning with five in-domain balloon throw trajectories,
4. **FIDO + Few-Shot Balloon**, which measures performance after fine-tuning with a combination of FIDO data and five in-domain balloon throw trajectories.

Due to the inference latency of the VLM (650ms on an L40S GPU), we fill our balloon with a mixture of helium and air to slow its fall (see Appendix C for experimental details). We conduct 50 trials of balloon catch for each method and record three outcomes: `Caught`, when the balloon

Fine-Tune Dataset	Robot Balloon Catching				Balloon Dynamics Modeling	
	Caught	Rim	Miss	Score	RMSE (↓)	DTW (↓)
Pretrain	12/50	27/50	11/50	25.5	1.28 ± 0.69	5.16 ± 1.91
FIDO	12/50	29/50	9/50	26.5	0.85 ± 0.26	3.35 ± 0.83
Few-Shot Balloon	12/50	31/50	7/50	27.5	0.81 ± 0.27	3.19 ± 0.97
FIDO + Few-Shot Balloon	**17/50**	**32/50**	**1/50**	**33.0**	$\mathbf{0.78 \pm 0.27}$	$\mathbf{3.11 \pm 0.93}$



Table 2. **Planning and offline evaluation with a FIDO Dynamics Model for Balloon Catch [Right].** We evaluate dynamics models fine-tuned on different data mixtures for the task of physical robot balloon catching (Caught / Rim / Miss) and on offline balloon dynamics modeling (RMSE / DTW). We consider catching success classified as either a successful catch, a rim hit, or a miss. We also calculate a composite score `#Caught` + $0.5\times$`#Rim`. Combining FIDO data with a small amount of balloon-specific data yields the strongest performance across both the physical catching task and offline trajectory prediction.

touches the bottom of the basket; `Rim`, when the balloon hits the rim of the basket before falling out; and `Miss`, when the balloon fails to make contact with any part of the basket. We report counts across these outcomes alongside a composite score `#Caught` + $0.5\times$`#Rim`.

Results. As shown in Table 2, fine-tuning on FIDO data alone yields modest improvements over the pretrained baseline in balloon catching, reducing misses from 11/50 to 9/50. We attribute the limited gains to the domain gap between fetch and balloon dynamics, as balloons are significantly more affected by air resistance and drift. Fine-tuning on a small amount of balloon-specific data similarly improves over the pretrained baseline (7/50 misses) but lacks the diversity to generalize across the range of trajectories encountered at test time. Combining FIDO data with the few-shot balloon data results in the strongest performance, catching 17/50 balloons and nearly eliminating misses (1/50), achieving the highest composite score across conditions.

We additionally perform offline evaluations on the 200 balloon trajectories recorded from the trials above, isolating dynamics-prediction quality from physical task execution. The offline results align with performance in balloon catching, where fine-tuning on a combination of FIDO and few-shot balloon data yields the best dynamics modeling. We believe the diverse fetch data provides a strong foundation for the general dynamics of falling objects under gravity, which the balloon-specific data then refines to align with this narrower domain. This complementary combination is enabled by FIDO's in-the-wild data collection strategy, which provides the breadth of physical conditions necessary to serve as a generalizable prior for downstream dynamics tasks.

6 Discussion and Conclusion

In this work, we present FIDO, a robot quadruped that collects grounded, in-the-wild dynamics data by playing fetch. By designing an interaction that is natural, repeatable, and accessible to non-expert participants, we enable data collection across diverse environments and a broad pool of interaction partners without specialized training or infrastructure. FIDO generalizes to two unseen domain-adjacent tasks of badminton shuttlecock and baseball ball trajectory prediction, achieving a **29.1%** and **26.8%** improvement over the pretrained baseline respectively, with performance improving monotonically with data diversity. FIDO's learned dynamics model is also directly useful for robot planning, enabling real-time balloon catching **20.0%** better than baselines trained without FIDO data. Together, these results suggest that the diversity afforded by in-the-wild data collection is central to generalizable physical dynamics learning.

Limitations and Future Work. While our results with $N = 18$ interaction partners across three environments are encouraging, scaling up data collection to more throwers, more environments, and longer deployment horizons would further stress-test the relationship between diversity and generalization. Additionally, our current data collection is limited to successful fetch interactions; incorporating failure modes such as missed catches and mid-air deflections, other multimodal interactions, and passive environmental observation could expose the model to a richer and more varied distribution of physical phenomena. We believe scaling up data collection could further unlock downstream applications such as reward learning and general-purpose video-based world modeling.

Acknowledgments

If a paper is accepted, the final camera-ready version will (and probably should) include acknowledgments. All acknowledgments go at the end of the paper, including thanks to reviewers who gave useful comments, to colleagues who contributed to the ideas, and to funding agencies and corporate sponsors that provided financial support.

References

[1] A. Mandlekar, J. Booher, M. Spero, A. Tung, A. Gupta, Y. Zhu, A. Garg, S. Savarese, and L. Fei-Fei. Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1048–1055. IEEE, 2019.

[2] S. Mirchandani, D. D. Yuan, K. Burns, M. S. Islam, T. Z. Zhao, C. Finn, and D. Sadigh. Robocrowd: Scaling robot data collection through crowdsourcing. In *International Conference on Robotics and Automation (ICRA)*, May 2025.

[3] Z. Sun and S. Song. Latent policy barrier: Learning robust visuomotor policies by staying in-distribution, 2026. URL https://arxiv.org/abs/2508.05941.

[4] A. Nagabandi, K. Konoglie, S. Levine, and V. Kumar. Deep Dynamics Models for Learning Dexterous Manipulation. In *Conference on Robot Learning (CoRL)*, 2019.

[5] C. Finn and S. Levine. Deep visual foresight for planning robot motion. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017.

[6] B. Ai, S. Tian, H. Shi, Y. Wang, T. Pfaff, C. Tan, H. I. Christensen, H. Su, J. Wu, and Y. Li. A review of learning-based dynamics models for robotic manipulation. *Science Robotics*, 10 (106):eadt1497, 2025. doi:10.1126/scirobotics.adt1497. URL https://www.science.org/doi/abs/10.1126/scirobotics.adt1497.

[7] Y. Wang, Y. Li, K. Driggs-Campbell, L. Fei-Fei, and J. Wu. Dynamic-Resolution Model Learning for Object Pile Manipulation. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi:10.15607/RSS.2023.XIX.047.

[8] D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint. Learning multi-object dynamics with compositional neural radiance fields. In *Proc. of the Annual Conf. on Robot Learning (CoRL)*, volume 205 of *Proceedings of Machine Learning Research*, pages 1755–1768. PMLR, 14–18 Dec 2023.

[9] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. *arXiv preprint arXiv:1811.04551*, 2018.

[10] J. Xu, E. Heiden, I. Akinola, D. Fox, M. Macklin, and Y. Narang. Neural robot dynamics. In *9th Annual Conference on Robot Learning*, 2025.

[11] Qwen Team. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[12] Gemini 2.5 Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

[13] M. J. Kim, Y. Gao, T.-Y. Lin, Y.-C. Lin, Y. Ge, G. Lam, P. Liang, S. Song, M.-Y. Liu, C. Finn, and J. Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. *arXiv preprint arXiv:2601.16163*, 2026.

[14] D. Tran, R. Martín-Martín, and K. Grauman. Egoexo-wm: Unlocking exo video for ego world models, 2026.

[15] Z. Zhu, X. Wang, W. Zhao, C. Min, N. Deng, M. Dou, Y. Wang, B. Shi, K. Wang, C. Zhang, Y. You, Z. Zhang, D. Zhao, L. Xiao, J. Zhao, J. Lu, and G. Huang. Is sora a world simulator? a comprehensive survey on general world models and beyond. *arXiv preprint arXiv:2405.03520*, 2024.

[16] A. Bardes, Q. Garrido, J. Ponce, M. Rabbat, Y. LeCun, M. Assran, and N. Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv:2404.08471*, 2024.

[17] Team Wan et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[18] T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, C. Ng, R. Wang, and A. Ramesh. Video generation models as world simulators, 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[20] K. Crowston. Amazon mechanical turk: A research tool for organizations and information systems scholars. In A. Bhattacherjee and B. Fitzgerald, editors, *Shaping the Future of ICT Research. Methods and Approaches*, pages 210–221. Springer Berlin Heidelberg, 2012.

[21] Prolific. Prolific. https://www.prolific.com, 2026. Online participant recruitment platform.

[22] R. Vaish, K. Wyngarden, J. Chen, B. Cheung, and M. S. Bernstein. Twitch crowdsourcing: Crowd contributions in short bursts of time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3645–3654. ACM, 2014.

[23] V. Liu, A. Adeniji, H. Zhan, R. Bhirangi, P. Abbeel, and L. Pinto. Egozero: Robot learning from smart glasses, 2025. URL https://arxiv.org/abs/2505.20290.

[24] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu. Egomimic: Scaling imitation learning via egocentric video, 2024. URL https://arxiv.org/abs/2410.24221.

[25] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[26] M. Du, J. Grannen, S. Song, and D. Sadigh. Will people enjoy a robot trainer? a case study with snoopie the pacerbot. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

[27] P. Liu, Y. Orru, C. Paxton, N. M. M. Shafiullah, and L. Pinto. Ok-robot: What really matters in integrating open-knowledge models for robotics. *arXiv preprint arXiv:2401.12202*, 2024.

[28] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *Conference on Robot Learning (CoRL)*, 2024.

[29] Physical Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

[30] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto. On bringing robots home. *arXiv preprint arXiv:2311.16098*, 2023.

[31] J. Grannen, M. Pan, K. Llontop, C. Ho, M. Zolotas, J. Bohg, and D. Sadigh. Robot-powered data flywheels: Deploying robots in the wild for continual data collection and foundation model adaptation, 2025. URL https://arxiv.org/abs/2511.19647.

[32] G. Jocher, J. Qiu, and A. Chaurasia. Ultralytics YOLO, Jan. 2023. URL https://github.com/ultralytics/ultralytics.

[33] U. Robotics. Sports services interfaces, 2026. URL https://support.unitree.com/home/en/developer/sports_services.

[34] L. Medeiros. Language segment-anything. https://github.com/luca-medeiros/lang-segment-anything, 2023.

[35] A. W. Harley, Y. You, X. Sun, Y. Zheng, N. Raghuraman, Y. Gu, S. Liang, W.-H. Chu, A. Dave, P. Tokmakov, S. You, R. Ambrus, K. Fragkiadaki, and L. J. Guibas. AllTracker: Efficient dense point tracking at high resolution. In *ICCV*, 2025.

[36] P. Li, T. Xiang, E. Mao, S. Wei, X. Chen, A. Masood, F.-F. Li, and E. Adeli. Quantiphy: A quantitative benchmark evaluating physical reasoning abilities of vision-language models. *arXiv preprint arXiv:2512.19526*, 2025.

[37] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[38] N.-E. Sun, Y.-C. Lin, S.-P. Chuang, T.-H. Hsu, D.-R. Yu, H.-Y. Chung, and T.-U. Ik. Tracknetv2: Efficient shuttlecock tracking network. In *Proceedings of the 2020 International Conference on Pervasive Artificial Intelligence (ICPAI)*. IEEE, 2020.

[39] A. Piergiovanni and M. S. Ryoo. Fine-grained activity recognition in baseball videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[40] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012. URL https://arxiv.org/abs/1212.0402.

[41] D. Friedman and A. B. Dieng. The vendi score: A diversity evaluation metric for machine learning, 2023. URL https://arxiv.org/abs/2210.02410.

[42] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. URL https://arxiv.org/abs/2304.07193.

FIDO the Robot Dog: Learning Dynamics Models via In-the-Wild Fetch Appendix

A FIDO Dataset Details

A.1 Data Processing

We collect dynamics data through the FIDO fetch interaction (§3): an human stands behind the robot and throws a ball which the robot retrieves and returns in a repeated interaction. Following collection, we post-process the video data to isolate the trajectory of the ball for each throw. The trajectory segments are trimmed through manual annotation, each beginning with the first video frame where the ball is visible and ending with the last frame before the robot begins moving to retrieve the ball. After trimming, the average trajectory clip length per throw is 1.64 seconds. We display sample trajectories from the FIDO dataset in Fig. 4.



Figure 4: **Sample FIDO trajectories.** We visualize one trajectory from each environment in the FIDO dataset. Sequential frames are overlaid one on top of another, with earlier frames displayed at a lower opacity than later frames to show the ball's movement over time.

A.2 Data Diversity

We quantify the diversity of the fetch datasets used for diversity ablations in Section 5 by measuring 1) diversity of the visual scene and 2) diversity of the ball trajectories. Each dataset contains 75 videos and is collected by a varying number of human throwers in a varying number of environments.

DINOv2 Vendi Score. To measure visual diversity of the videos, we calculate the Vendi Score [41] over DINOv2 [42] image embeddings within each dataset. From each video we sample frames at a fixed stride, embed every frame with a DINOv2 model, and perform an L2-normalization on each embedding. The Vendi Score is then computed over these embeddings and can be interpreted as the effective number of distinct frames in the dataset. Higher scores indicate greater within-dataset variation in visual appearance.

Within-group DTW distance. To measure diversity of ball trajectories, we compute Dynamic Time Warping (DTW) distances between pairs of trajectories within each dataset. From each video we extract a 2D trajectory and remove translational shift by subtracting the starting position of the ball. We then compute the DTW distance between every pair of unique trajectories, resulting in a measure of within-dataset trajectory shape variation.

Num. Throwers	Num. Envs	DINOv2 Vendi Score	Within-group DTW
2	1	3.79 ± 0.16	0.617 ± 0.051
2	3	6.22 ± 0.29	0.560 ± 0.047
18	3	7.03 ± 0.35	0.769 ± 0.067

Table 3: **FIDO diversity ablation statistics.** We quantify the visual and trajectory diversity of each dataset used for diversity ablations in Section 5. We measure visual diversity using the Vendi Score computed on DINOv2 embeddings, and measure trajectory diversity using within-group DTW distances. The FIDO dataset is the last row (18 throwers, three environments) and is the most diverse by both measures.

A.3 Participant Privacy

The FIDO system captures forward-facing video, which excludes the interaction partner in the collected ball trajectories. As data collection scales, protecting participant privacy becomes an increasingly important consideration. Future safeguards could include careful handling of incidentally captured bystanders through automated data anonymization.

B Additional VLM Dynamics Model Details

B.1 Training Details

We provide additional training details for the VLM dynamics models used throughout the paper. Across experiments, we use a common image-conditioned trajectory-prediction interface built on Qwen3-VL-2B-Instruct, while varying the source of supervision. Our comparisons include the pretrained backbone, adapted models trained on FIDO data, and balloon-specific variants trained on balloon supervision alone or on mixtures of FIDO and balloon data. What remains fixed across settings is the model interface introduced in Section 4: each example consists of a single RGB image from the first image of the observation window, five observed object positions, and a future prediction query.

To construct supervision, we begin from tracked trajectories and retain contiguous visible segments in which the object is present. From each valid segment, we sample windows with a history of five observed points and a future prediction query. Coordinates are normalized from the original image frame resolution, typically 1280×720, into integer coordinates in $[0, 999]$, and the future target is represented as semicolon-separated (x, y) pairs.

B.2 Prompts

We now describe the prompts we use in our VLM instantiation of a dynamics model. The prompt for single-stride prediction examples has the form:

Single stride prediction prompt

```
<image>
Ball trajectory in a video. The image shows frame 1.
Coordinates are integers in [0, 999] where (0,0) is top-left.

Context (frame: x,y):
Frame 1: x1,y1
Frame 2: x2,y2
Frame 3: x3,y3
Frame 4: x4,y4
Frame 5: x5,y5

Predict the ball at the next K consecutive frames (frames 6 to 5+K).
Output ONLY as K semicolon-separated integer pairs:
x1,y1;x2,y2;...;xK,yK
Do NOT output any other text.
```

We use a separate prompt for multi-stride predictions where the future query explicitly asks for predictions at a coarser temporal stride. In that case, the prompt changes only in the final instruction:

Multi-stride prediction prompt

```
<image>
Ball trajectory in a video. The image shows frame 1.
Coordinates are integers in [0, 999] where (0,0) is top-left.

Context (frame: x,y):
Frame 1: x1,y1
Frame 2: x2,y2
Frame 3: x3,y3
Frame 4: x4,y4
Frame 5: x5,y5

Predict the ball at the next K frames sampled every S frames
(frames 6, 6+S, ..., 6+(K-1)S).
Output ONLY as K semicolon-separated integer pairs:
x1,y1;x2,y2;...;xK,yK
Do NOT output any other text.
```

B.3 Dataset Processing

Fetch data. We use the pre-fetch portion of each robot-collected video: the segment from when the ball first becomes visible in the robot camera until before the robot starts moving to retrieve it. We retrieve ball trajectories in image space using LangSAM + AllTracker pipeline — LangSAM is used to identify the first frame in which the ball is visible and AllTracker tracks the ball center across subsequent frames. For training, we first sample and process 75 ball trajectories from each training subset, and sample multiple training windows from each contiguous ball trajectory.

Balloon data. For balloon supervision, we first track the balloon in image space using hardcoded HSV range for the balloon color. The balloon in our videos falls slower than the ball. To account for this, for our balloon finetuning or co-finetuning experiments, we use a temporally downsampled balloon dataset in which every fourth frame of the original video is retained.

Badminton data. We evaluate on processed TrackNetV2 [38] trajectories derived from the official dataset annotations. TrackNetV2 provides frame-level shuttlecock locations, which we convert into trajectory prediction interface using the same interface as our FIDO data. The original dataset includes multiple rallies, each comprising multiple back-and-forth shots between the players. Starting from the annotations, we detect candidate hit frames from changes in shuttlecock motion, using acceleration and direction change cues together, while using a post-hit speed threshold to avoid splitting at natural arc peaks. We then discard segments with insufficient length, visibility, or motion, and perform a manual review pass to discard bad segments. From the remaining clips, we sample 100 for evaluations.

Baseball data. For baseball, we evaluate on a filtered subset of 77 baseball pitches from MLB YouTube clips [39]. Unlike badminton data, the dataset did not directly provide frame-level annotations for the pixel locations of the ball, so we generate these ourselves. The dataset does provide annotations on time intervals for pitches within baseball gameplay videos. We use them to first extract pitch-centered clips and then use LangSAM [23] to produce an initial ball localization. These detections seed AllTracker [35] which propagates the ball across the clip and outputs a per-frame image-space trajectory. We then apply several stages of filtering to remove failed LangSAM detections, invalid tracks, obvious non-pitch motion, extremely small motion, return throw motion or spirals. From the remaining clips, we retain a manually selected subset of 77 high-quality pitches for evaluation, where the ball remains visually consistent and corresponds to a single pitch motion.

B.4 Optimization Hyperparameters

We use a common LoRA finetuning recipe across our dynamics-models experiments. The model is trained on trajectory-prediction examples constructed from fetch and/or balloon data, using the shared prompt format described above. For the balloon cotraining experiments, we sample an equal number of fetch and balloon training samples at a balanced $50{:}50$ ratio.

The Fetch only variants in Table 1 use a single-stride variant of this same training setup. In those runs, the future stride is fixed to 1, prediction lengths are sampled from $\{4, 8\}$ and the models are trained for 1000 effective gradient steps.

Hyperparameter	Value
Backbone model size	2B
LoRA rank r	8
LoRA α	16
LoRA dropout	0.05
Optimizer	AdamW
Base Learning rate	1×10^{-5}
Learning-rate schedule	cosine decay
Warmup ratio	0.03
Training image resolution	320×180
Effective batch size	8
Maximum generation length	96 tokens
Effective training steps	1500
Maximum predicted points	32

Table 4: **Hyperparameters**. Optimization hyperparameters used for training the dynamics models in Table 2.

B.5 Evaluation Metrics

In our quantitative evaluations, we report root mean square error (RMSE) and dynamic time warping (DTW), both normalized by the ground truth trajectory length. We report mean and 95% confidence intervals over evaluation samples, computed as 1.96 times the standard error of the sample mean. For an evaluation sample, let $\hat{Y}_t = \{\hat{p}_{t+\delta_1}, \ldots, \hat{p}_{t+\delta_K}\}$ denote a predicted trajectory and let $Y_t = \{p_{t+\delta_1}, \ldots, p_{t+\delta_K}\}$ denote the corresponding ground truth trajectory.

Normalization. We normalize all metrics by the geodesic distance traversed by the ground truth trajectory. The normalizing length is given by: $L(Y_t) = \sum_{i=1}^{K-1} \|p_{t+\delta_{i+1}} - p_{t+\delta_i}\|_2$.

RMSE: Normalized RMSE measures the normalized time-aligned Euclidean deviation between predicted and ground truth future coordinates, scaled by total length of the trajectory:

$$\mathrm{RMSE}(\hat{Y}_t, Y_t) = \frac{\sqrt{\frac{1}{K}\sum_{i=1}^{K} \|\hat{p}_{t+\delta_i} - p_{t+\delta_i}\|_2^2}}{L(Y_t)}$$

DTW: Standard DTW measures trajectory similarity while allowing non-linear alignment along time, thus making it less sensitive than RMSE to small temporal offsets between ground truth and predicted motion. We report DTW normalized by the trajectory length $L(Y_t)$.

C Additional Experimental Details

C.1 Baseline Implementations

Classical predictor. We tune a family of classical trajectory predictors using FIDO trajectories, including constant-velocity and constant-acceleration Kalman filters, least-squares linear and quadratic trajectory fits, particle filters, and variants that blend their predictions with the last observed position. We sweep $14{,}056$ configurations over predictor type, process and observation noise,

acceleration damping, and the weighting of the last-position prediction, and select the variant with the lowest RMSE.

Coordinates-only LLM. We fine-tune the same model described in Section B on FIDO trajectories while removing the image input, leaving only the trajectory history H_t and timestep query Δ as inputs.

Internet-tuned VLM. We fine-tune the same model described in Section B on projectile trajectories extracted from internet videos. We collect candidate projectile-motion clips from UCF-101 [40] and process them using the same trajectory extraction pipeline applied to the MLB dataset, discarding clips with tracking failures or unsuitable motion. From the remaining clips, we sample 75 trajectories to match the size of the FIDO dataset.

C.2 Balloon Catching Experiment Setup

We evaluate our learned dynamics model on the task of catching a thrown balloon in a basket attached to the head of the robot. The robot begins on the left side of the balloon arena, a rectangular area of 2.4×1.4 meters. The balloon is thrown from a fixed position on the right side of the arena towards the center, and the robot begins to move once the balloon crosses the rightmost wall. Throughout the trial, the robot must stay within the boundary of the arena as it attempts to catch the balloon.

Due to the inference latency of our dynamics model, we fill our balloon with a mixture of helium and air to slow its fall. We standardize the buoyancy of the balloon across trials by calibrating the mixture such that the balloon takes 4 seconds to fall to the ground from a height of 1.5 meters.

Given the dynamics model's trajectory predictions, we leverage known camera extrinsics and an Aruco tag mounted on the back of the robot to determine the waypoints for the robot to execute, computed as a linear trajectory towards the predicted future location of the balloon.

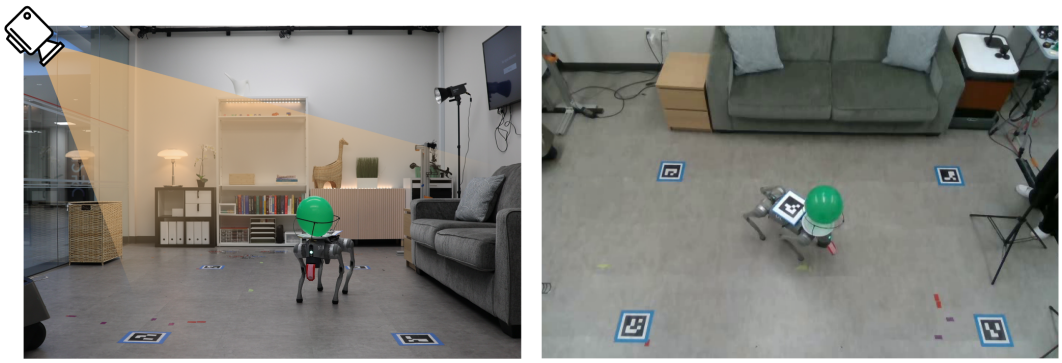


Figure 5: **Balloon catching setup.** We mount an Intel RealSense D435i at a downward angle toward the balloon arena [**Left**]. In the camera view, four Aruco tags on the floor mark the corners of the balloon arena and one additional Aruco tag localizes the position of the robot [**Right**].