Overcoming Distribution Shifts with Autonomous Embodied Data Collection

Anonymous Submission

***Abstract*—Distribution shifts are a fundamental challenge in machine learning that can significantly limit model deployment, especially in robotics where models must handle messy real-world scenarios. The most direct way to overcome these shifts is to collect additional data suited for the deployment domain. However, collecting this data manually can be expensive and difficult to specify. In this work, we propose Autonomous Embodied Data Adaptation (AEDA), where we instead leverage autonomous robotic systems themselves to collect data targeted at overcoming distribution shifts. AEDA uses large multimodal models (LMMs) to identify distribution shifts, construct data collection plans to address them, and execute these plans on a robot. We instantiate AEDA on a real-world mobile manipulator to improve depth estimation of a pretrained foundation model, and show that its data improves prediction accuracy by 17.5% overall compared to a baseline. Additional videos and details: https://robot-data-collector.github.io**

***Index Terms*—Domain Adaptation, Active Data Collection**

I. Introduction

Distribution shifts between training data and real-world deployment remain a pervasive challenge in machine learning [1]. While well-studied in traditional supervised settings [2], [3], this problem takes on new dimensions when we have the opportunity to bridge these shifts with additional data collection [4]. However, doing this manually can be expensive and require significant domain-specific considerations. Furthermore, it is often unclear *what* data is most valuable to collect, and ensuring that human collectors produce the right data adds further inefficiency [5], [6].

This problem is especially acute for foundation models, particularly large multimodal models (LMMs), when applied to robotics, where physical and spatial reasoning capabilities are essential. While LMMs have demonstrated strong generalization, their pre-training on curated, internet-sourced data leaves them surprisingly brittle on tasks requiring precise spatial understanding, such as depth estimation, object localization, and scene geometry reasoning. This limitation persists even in models explicitly designed for robotics applications: for instance, Gemini Robotics-ER [7], [8], a state-of-the-art closed-source model trained with embodied reasoning in mind.

The dominant remedy to patch these gaps is to collect additional domain-specific data [9], [10]. However, this targeted collection is typically done by hand, resulting in the cost and targeting challenges described above. A natural question thus arises: can we instead use robotics itself to make LMMs more effective for robotics?

We propose that autonomous robotic systems can serve as active data collectors for LMMs, targeting collection toward the specific domain gaps between their training data and deployment in robotics settings. As robots become increasingly capable of operating in the real world, they offer a natural platform for acquiring physically-grounded data at scale, without the overhead of human supervision.

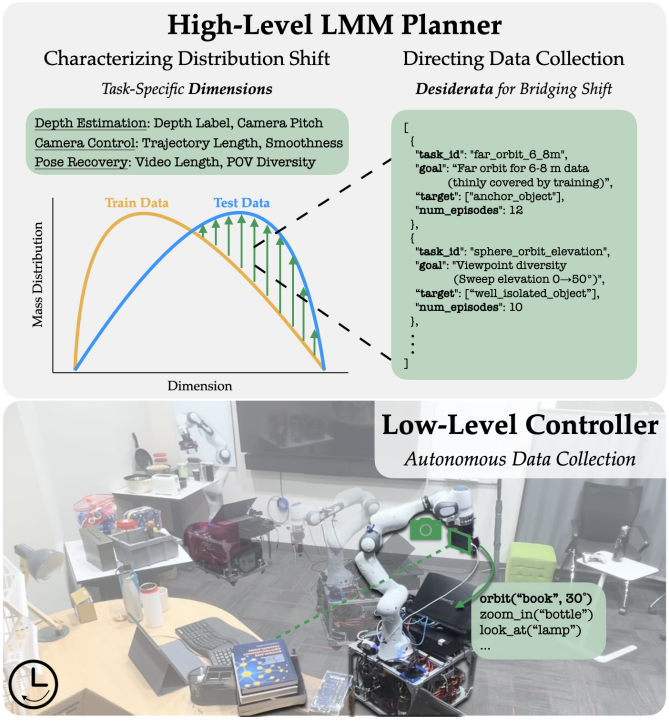

Fig. 1: **Leveraging robots as active data collectors to target distribution shifts between training and deployment (test) settings:** We propose Autonomous Embodied Data Adaptation (AEDA), a framework for autonomous embodied data collection that actively reduces distribution shift. AEDA is comprised of a high-level LMM Planner [**Top**], which characterizes distribution shift across self-proposed dimensions, as well as generates a set of desiderata to direct autonomous data collection. This desiderata then gets passed a low-level Code-As-Policies controller [**Bottom**], which generates code to execute autonomous data collection over multiple hours.

Concretely, we propose a three-stage framework. First, an LMM characterizes the domain gap between existing training data and samples from the target deployment domain by identifying *dimensions* of variation along which the distributions diverge. Second, the LMM synthesizes this characterization into a data collection plan – a set of structured *desiderata* specifying what data to gather. Third, we employ a Code-as-Policies (CaP) controller [11], [12], in which the LMM writes code to command a robot, to autonomously execute the collection plan. Crucially, the spatial reasoning capabilities required to execute a data collection plan are substantially simpler than those downstream reasoning tasks we aim to improve, making

this a tractable use of current foundation model capabilities. We instantiate AEDA on a mobile manipulator, targeting depth prediction as a representative spatial reasoning capability.

Our contributions are: **(1)** Autonomous Embodied Data Adaptation (AEDA), a framework for autonomous embodied data collection which uses a LMM to characterize distribution shifts and control a robot to collect targeted data to address them, and **(2)** an instantiation of this framework for improving monocular depth estimation capabilities in a pretrained foundation model on two distinct target domains. We run two independent data collection campaigns (one per target domain), and show that fine-tuning on this targeted data yields an overall **17.5%** improvement in depth prediction accuracy over the baseline, demonstrating that autonomous embodied collection can be an effective and efficient strategy for overcoming distribution shifts.

II. Related Work

Our approach sits at the intersection of domain adaptation and autonomous data collection.

A. Active Domain Adaptation

A large body of work addresses distribution shift from a statistical perspective, characterizing it via measures such as KL divergence, maximum mean discrepancy (MMD) [13], or density ratio estimation [14]. While these methods offer strong theoretical guarantees, they often rely on restrictive parametric assumptions and can be difficult to apply in high-dimensional, real-world settings. Our work instead leverages the semantic reasoning capabilities of LMMs to characterize distribution gaps [15], complementing these statistical tools with more practical, open-ended characterizations.

In settings where additional data collection is infeasible, a line of work considers test-time adaptation (TTA): adapting a model at inference time via auxiliary task prediction, distribution alignment of test samples, or prompt tuning [16]–[18]. These methods are constrained to work with whatever the test sample provides, without the ability to actively reshape it.

When new data *can* be collected, prior work has studied how to ensure that collection targets the deployment distribution. DAgger [19] and its variants [20] address covariate shift in imitation learning such that the training data matches the state distribution induced by the learned policy. Gandhi et al. [6] studies how to elicit human-provided data to improve model capabilities. Our work shares the goal of targeted collection, but replaces human collectors with an autonomous robot.

B. Autonomous Data Collection

The question of how an autonomous agent should explore to maximize learning is central to reinforcement learning (RL). A rich literature studies intrinsic motivation and exploration bonuses, rewarding agents for visiting novel states [21], [22]. Our work is related in spirit: we seek to direct autonomous behavior toward data that is informative for improving a model. However, rather than optimizing for novelty within an RL policy, we use an LMM planner to explicitly reason about what data is missing and direct collection accordingly.

Most related to our approach is the emerging paradigm of robot-powered data flywheels [23], [24], where deployed robots act as both task executors and data generators. Our work is complementary to this: while data flywheel approaches focus on large-scale autonomous deployment to broadly improve data coverage, we focus on *targeted* collection that explicitly closes a *diagnosed* distribution gap, using LMMs not just to execute collection, but to first reason about *what* to collect.

III. Autonomous Embodied Data Adaption

AEDA consists of three stages: characterizing the gap between training and target domains, generating a plan for active data collection, and autonomously executing that plan.

A. Characterizing Distribution Shift with a LMM

First, AEDA characterizes the gap between the training and target distributions. Given samples $\mathcal{D}_{\text{train}} = \{x_i\}_{i=1}^{n} \sim P_{\text{train}}$ and $\mathcal{D}_{\text{target}} = \{x_j\}_{j=1}^{m} \sim P_{\text{target}}$, we prompt an LMM to identify dimensions along which they differ. Concretely, the LMM proposes a set of K measurable functions $\{f_k\}_{k=1}^{K}$, where each $f_k : \mathcal{X} \to \mathbb{R}$ maps a sample to a scalar value along an interpretable axis of variation (e.g. object distance or size). The discrepancy along each dimension is then evaluated as the W1 Wasserstein distance: $\Delta_k = W_1\big(f_k(\mathcal{D}_{\text{train}}),\ f_k(\mathcal{D}_{\text{target}})\big)$. The dimension is retained if a meaningful gap is observed, i.e., $\Delta_k > \epsilon$ and passed to the next stage as a structured characterization of the domain gap.

B. Desiderata for Directing Data Collection

The LMM converts the retained dimensions $\{f_k\}$ into a set of "desiderata": a list of collection plans structured as a JSON of tasks. Each task specifies a target object class, a required scene, predominantly arm- or base-driven motions, a trajectory drawn from the primitive catalog, a collection episode budget, and per-episode acceptance criteria. The primitive catalog contains 7 trajectory families, spanning: sphere orbit, far orbit, approach and retreat, linear, dolly, and look-away-return. This sits at the *task* level, abstract enough that the LMM can plan without solving geometry, yet concrete enough that each entry maps unambiguously to a callable skill.

The prompt consists of a fixed preamble defining the LMM's role, the output schema, and the primitive catalog with associated reach envelopes, followed by the ranked per-dimension discrepancies Δ_k, aggregate dataset statistics, and optional soft guidance from a human operator. Episode budgets are allocated in rough proportion to Δ_k, so that larger distribution gaps receive more collection effort. Importantly, no per-task few-shot examples or prompt tuning are required — only the platform capability description needs to change across different robot embodiments.

C. Robot Controller for Executing Data Collection

At execution time, a LMM orchestrates a catalog of $\sim$25 predefined parametrized robot skills to execute the collection plan generated in the previous stage. The controller operates at three nested timescales — task, episode, and control rate —

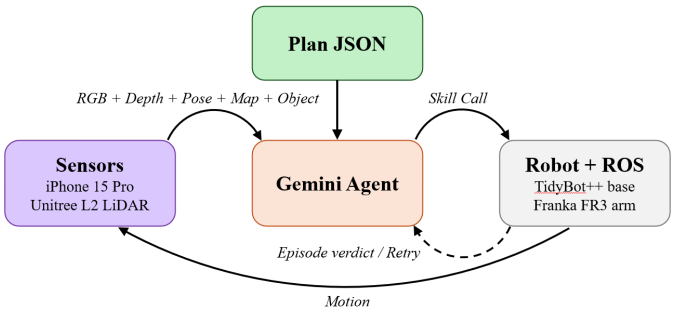

Fig. 2: **Robot Controller.** The LMM controller executes each task by selecting skills from a predefined catalog. Autonomous execution is maintained through closed-loop feedback at three timescales: low-level MPC replanning against a rolling costmap, per-episode acceptance evaluation, and task-level budgeting that retries or advances based on collected episode count.

each with its own feedback loop, enabling closed-loop, fully autonomous execution without in-context demonstrations.

Skill Catalog. Skills span four categories. First, perception skills include scene survey, open-vocabulary object listing, and target detection via a RAM++ [25] to SAM3 [26] cascade. Second, a set of navigation skills expose goal-driven Nav2 [27] planning and frontier-based exploration. For data collection, an atomic *capture-episode* skill bundles trajectory generation, execution, and per-waypoint recording. Finally, a set of arm movement skills consider resets to home and target-height poses. The agent consumes one task at a time with a compact primer prompt that describes the skill catalog and task-conditional rules. Past context and history persists across tasks, preventing already-visited objects from being recollected.

Trajectory Execution and Safety. Each *capture-episode* call begins with a cost-aware feasibility check against the fused SLAM and Nav2 rolling costmaps, rejecting base trajectories that pass through lethal cells. Orbit-shaped trajectories are additionally refined by a Nelder-Mead search [28] optimizing for reachability and data diversity. Arm trajectories are resolved through per-waypoint inverse kinematics, tagging infeasible waypoints rather than aborting. Base trajectories are tracked by a model-predictive controller [29] that re-optimizes the remaining arc mid-execution whenever the costmap reveals a newly-occupied cell ahead.

Closed-Loop Execution. As illustrated in Figure 2, the system closes the loop at three timescales: at the *control rate*, via continuous replanning of base trajectories against the costmap; at the *episode level*, via acceptance evaluation that returns a verdict to the agent, allowing it to retry with adjusted parameters; and at the *task level*, via budgeting, where the runner advances to the next task only after the required number of accepted episodes or attempt budget is reached.

IV. Experiments

To instantiate AEDA, we use a Franka FR3 [30] arm mounted on a TidyBot++ [31] mobile base. The system is equipped with a base-mounted Unitree L2 LiDAR for localization and mapping, and an iPhone 15 Pro wrist camera. The robot is controlled via the Gemini 3 Flash [32] with a set of predefined parametrized skills. We note that the robot is initialized in environments that broadly satisfy its self-proposed desiderata's scene requirements, for example access to larger rooms with space to maneuver or many relatively small objects to observe. We then let the robot roll out autonomously within these scenes.

To evaluate the efficacy of AEDA's data collection, we consider the task of monocular depth estimation with a pre-trained foundation model LMM. We select the Qwen2.5-VL-7B-Instruct [33] model as our base model and use three depth datasets with semantic annotations for fine-tuning and evaluation: SUN RGB-D [34] as our train domain, and NYU Depth V2 [35] and ARKitScenes [36] as our target settings.

Characterizing Distribution Shifts. As in Section III-A, AEDA first characterizes distribution shifts between train and target domains along self-proposed dimensions – "distance" and "object size" in this task of monocular depth estimation. We visualize the results of the "distance" characterization in Figure 3, Left. For each domain, we randomly sample 10% from the training split. To avoid potential bias, we evaluate exclusively on test splits, ensuring no overlap between the data used for collection planning and evaluation.

As shown in Figure 3, Left, we note that the actively-collected datasets exhibit markedly different depth distributions from the SUN RGB-D training set, confirming the domain gaps identified by the LMM. ARKit-targeted collection is heavily concentrated in the near range, with a sharp peak below 1m, reflecting the LMM's diagnosis that ARKit scenes contain a higher proportion of close-range objects than the training set. NYU-targeted collection, by contrast, is more spread across mid-to-far ranges, with a notable long tail extending beyond 5m — a region that is substantially underrepresented in SUN RGB-D. Importantly, the pre-existing ARKitScenes and NYU Depth V2 evaluation distributions differ meaningfully from SUN RGB-D in precisely these respects, validating that the LMM-proposed collection directions target genuine and consequential distribution gaps.

Model	ARKit [36]	NYU Depth V2 [35]
Pre-Trained Qwen2.5	1.121 ± 1.044	1.562 ± 1.401
Pre-Trained Gemini ER	0.459 ± 0.378	0.587 ± 0.561
FT: Train Set	0.595 ± 0.395	0.494 ± 0.637
FT: Train Set + Passive Data	0.426 ± 0.363	0.559 ± 1.602
FT: Train Set + Active Data (AEDA)	$\mathbf{0.417 \pm 0.431}$	$\mathbf{0.469 \pm 0.526}$

Table 1: **Depth Estimation with AEDA:** We report mean absolute error and standard deviation of predictions on two target domains.

Improving Models with Active Data Collection. Given an autonomously-collected dataset, we investigate how passive versus active data collection strategies affect model performance. We fine-tune a VLM (Qwen2.5-VL-7B-Instruct [33]), pre-trained on internet data, on three different data mixtures: **(1) Train Set**, in this case the Sun RGB-D dataset; **(2) Train Set + Passive Collection**, which incorporates autonomously-collected data *not* actively targeted at bridging the domain gap of the target domain; and **(3) Train Set + Active Collection (AEDA)**, which incorporates our actively-collected data in addition to the initial train set. For efficiency, we operationalize

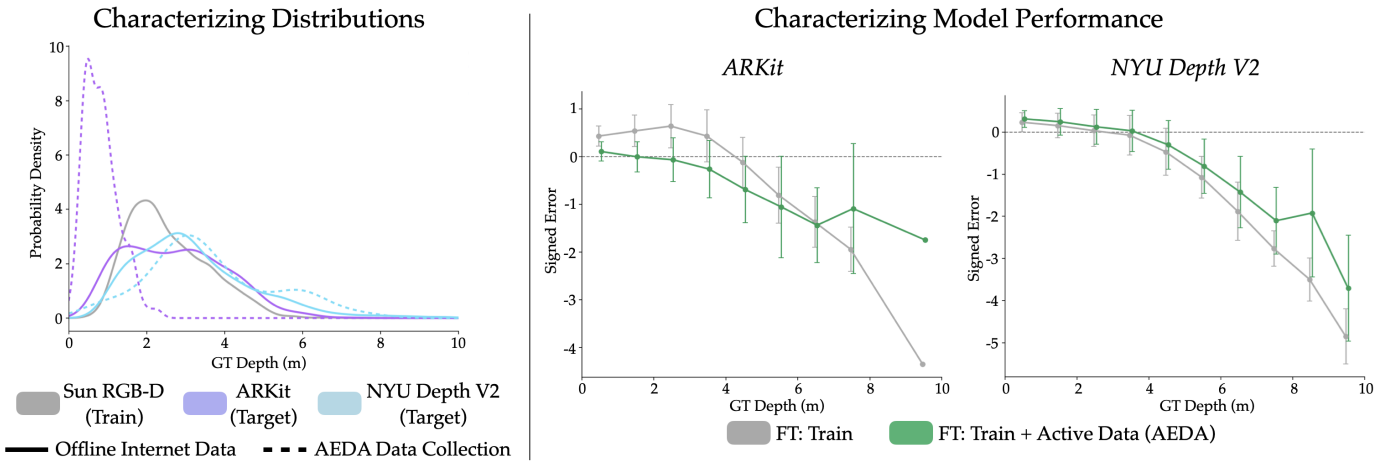

Fig. 3: **Characterizing Data Distributions and Model Performance.** Our LMM planner first characterizes samples from each data domain along self-proposed dimensions of variance (i.e. depth), and then directs data collection to patch these gaps [**Left**]. We also visualize model performance relative to this dimension both with and without AEDA's actively collected data.

"passive collection" as the data collected to bridge the *other* target domain's gap, as it is autonomously collected but untargeted with respect to the domain being evaluated.

Table 1 presents mean absolute error and standard deviation of model predictions across all conditions. Fine-tuning with actively-collected data yields the best results overall, with improvements of 29.9% and 5.06% over the train-set-only baseline across the ARKit and NYU domains respectively.

Interestingly, passive data collection also yields a nontrivial improvement on the ARKit evaluation set. We hypothesize this is because any out-of-distribution data broadens the diversity of the training mixture, improving model robustness even without explicit targeting. In support of this, Figure 3, Left shows that the NYU-targeted collection — which serves as our passive baseline for ARKit — has a long tail that partially overlaps with the ARKit domain, providing incidental but nontrivial coverage. We do not observe the same effect on NYU, which we attribute to the high specificity of the data collected when targeting ARKit: unlike truly random collection, it is narrowly distributed around the ARKit domain and thus provides little coverage of NYU's distribution. We hypothesize that a truly passive, domain-agnostic collection baseline would yield a larger improvement on NYU. We also compare against the pre-trained Gemini Robotics-ER 1.6 [7], highlighting the difficulty of spatial reasoning tasks such as depth estimation even for state-of-the-art closed-source foundation models explicitly designed for robotics applications.

We additionally visualize model failure modes in Figure 3, Right stratified by depth — one of the dimensions proposed by the LMM for characterizing domain gaps. For ARKit, we observe substantial performance gains in near-range settings, driven by the large proportion of short-range data in the actively-collected set: 90.6% of ARKit-targeted samples lie within 1.5m, compared to only 14.4% of the SUN RGB-D training set. For NYU Depth V2, improvements are concentrated at far-range distances, where 22.6% of the newly collected data lies beyond 5m versus only 1.5% in SUN RGB-D — directly reflecting the distribution gap identified during planning. We also observe a slight degradation in near-object estimation on NYU, which we attribute to the limited object diversity and low proportion of near-range samples in the NYU-targeted collection, which shifts the overall training distribution toward farther depths.

Autonomous Embodied Data Collection. We deployed the robot in two scenes, one for each target domain, according to the LMM desiderata directing the data collection. For the ARKit active data collection, the robot operated in an office setting for 61 minutes with zero human interventions. The robot collects a total of 6,852 image frames, which are filtered during post-processing to 1,293 frames containing 2,632 total object instances. The resulting dataset accounts for 7.61% of the final training mixture.

For the NYU active data collection, the robot was deployed in a large conference room setting for 58 minutes, collecting 10,228 image frames. After post-processing, 1,109 frames were retained with 1,367 object instances. The resulting dataset accounts for 4.10% of the final training mixture. This deployment required a single human intervention to reposition the robot and enable exploration of previously unseen areas.

V. Conclusion

We introduce AEDA, a framework that leverages an LMM to characterize distribution gaps between training and deployment data, then directs a robot to autonomously collect targeted data to bridge them. Applied to monocular depth estimation, fine-tuning on AEDA's actively collected data yields a **17.5%** improvement in VLM performance over fine-tuning on the original training set alone.

Limitations and Future Work. While AEDA operates autonomously during data collection, it currently requires occasional high-level supervision to transition between scenes and settings. A related open question is how to incentivize exploration for data diversity beyond the LMM-proposed dimensions, particularly in static environments. As embodied autonomy capabilities continue to improve, we look forward to extending AEDA to broader deployment settings, enabling robots to roam more freely and collect richer, more diverse data with minimal oversight.

REFERENCES

[1] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, T. Lee, E. David, I. Stavness, W. Guo, B. A. Earnshaw, I. S. Haque, S. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang, "Wilds: A benchmark of in-the-wild distribution shifts," in *International Conference on Machine Learning*, 2021.

[2] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, Eds., *Dataset Shift in Machine Learning*. MIT Press, 2009.

[3] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, 2000.

[4] V. Prabhu, A. Chandrasekaran, K. Saenko, and J. Hoffman, "Active domain adaptation via clustering uncertainty-weighted embeddings," in *International Conference on Computer Vision*, 2021.

[5] R. Mahmood, J. Lucas, J. M. Alvarez, S. Fidler, and M. T. Law, "Optimizing data collection for machine learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, November 2022.

[6] K. Gandhi, S. Karamcheti, M. Liao, and D. Sadigh, "Eliciting compatible demonstrations for multi-human imitation learning," in *Conference on Robot Learning*, 2022.

[7] Google DeepMind, "Gemini robotics-ER 1.6: Enhanced embodied reasoning," https://deepmind.google/blog/gemini-robotics-er-1-6/, Apr. 2026, aPI model ID: `gemini-robotics-er-1.6-preview`.

[8] G. R. Team, "Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer," *arXiv preprint arXiv:2510.03342*, 2025.

[9] Z. Ke, Y. Ming, and S. Joty, "Adaptation of large language models," in *Proceedings of the 2025 Annual Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 5: Tutorial Abstracts)*, 2025, pp. 30–37.

[10] A. Miyai, J. Yang, J. Zhang, Y. Ming, Y. Lin, Q. Yu, G. Irie, S. Joty, Y. Li, H. Li, Z. Liu, T. Yamasaki, and K. Aizawa, "Generalized out-of-distribution detection and beyond in vision language model era: A survey," *Transactions on Machine Learning Research*, 2025.

[11] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International conference on robotics and automation (ICRA)*. IEEE, 2023, pp. 9493–9500.

[12] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," *arXiv preprint arXiv:2209.11302*, 2022.

[13] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. 25, pp. 723–773, 2012. [Online]. Available: http://jmlr.org/papers/v13/gretton12a.html

[14] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Advances in neural information processing systems*, 2008, pp. 1433–1440.

[15] J. Gao, D. Sadigh, S. Huang, and D. Shah, "Grounding robot generalization in training data via retrieval-augmented vlms," *arXiv preprint arXiv:2603.11426*, 2026.

[16] S. Niu, J. Wu, Y. Zhang, Y. Chen, S. Zheng, P. Zhao, and M. Tan, "Efficient test-time model adaptation without forgetting," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 17–23 Jul 2022, pp. 16 888–16 905.

[17] J. A. Samadh, M. H. Gani, N. Hussein, M. U. Khattak, M. M. Naseer, F. S. Khan, and S. Khan, "Align your prompts: Test-time prompting with distribution alignment for zero-shot generalization," in *Conference on Neural Information Processing Systems*, 2023.

[18] Q. Wang, O. Fink, L. Van Gool, and D. Dai, "Continual test-time domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 7201–7211.

[19] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. PMLR, 11–13 Apr 2011, pp. 627–635.

[20] J. Zhang and K. Cho, "Query-efficient imitation learning for end-to-end autonomous driving," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017.

[21] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *ICML*, 2017.

[22] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros, "Large-scale study of curiosity-driven learning," in *ICLR*, 2019.

[23] 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," *arXiv preprint arXiv:2511.19647*, 2025.

[24] M. Ahn, D. Dwibedi, C. Finn, M. G. Arenas, K. Gopalakrishnan, K. Hausman, B. Ichter, A. Irpan, N. Joshi, R. Julian, S. Kirmani, I. Leal, E. Lee, S. Levine, Y. Lu, I. Leal, S. Maddineni, K. Rao, D. Sadigh, P. Sanketi, P. Sermanet, Q. Vuong, S. Welker, F. Xia, T. Xiao, P. Xu, S. Xu, and Z. Xu, "Autort: Embodied foundation models for large scale orchestration of robotic agents," 2024.

[25] Y. Zhang, X. Huang, J. Ma, Z. Li, Z. Luo, Y. Xie, Y. Qin, T. Luo, Y. Li, S. Liu, Y. Guo, and L. Zhang, "Recognize anything: A strong image tagging model," *arXiv preprint arXiv:2306.03514*, 2023.

[26] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, J. Lei, T. Ma, B. Guo, A. Kalla, M. Marks, J. Greer, M. Wang, P. Sun, R. Rädle, T. Afouras, E. Mavroudi, K. Xu, T.-H. Wu, Y. Zhou, L. Momeni, R. Hazra, S. Ding, S. Vaze, F. Porcher, F. Li, S. Li, A. Kamath, H. K. Cheng, P. Dollár, N. Ravi, K. Saenko, P. Zhang, and C. Feichtenhofer, "SAM 3: Segment anything with concepts," https://arxiv.org/abs/2511.16719, 2025.

[27] S. Macenski, F. Martin, R. White, and J. Ginés Clavero, "The marathon 2: A navigation system," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[28] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[29] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[30] Franka Robotics, "Franka research 3," https://franka.de/franka-research-3, 2023.

[31] J. Wu, W. Chong, R. Holmberg, A. Prasad, Y. Gao, O. Khatib, S. Song, S. Rusinkiewicz, and J. Bohg, "Tidybot++: An open-source holonomic mobile manipulator for robot learning," in *Conference on Robot Learning*, 2024.

[32] Google DeepMind, "Gemini 3 flash preview," https://ai.google.dev/gemini-api/docs/models/gemini-3-flash-preview, 2025.

[33] Qwen Team, "Qwen2.5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025.

[34] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *CVPR*, 2015.

[35] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[36] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Nießner, "ARKitScenes: A diverse real-world dataset for 3d indoor scene understanding," in *NeurIPS Datasets and Benchmarks Track*, 2021.