A System-Level View on Out-of-Distribution Data in Robotics

Rohan Sinha, Apoorva Sharma, Somrita Banerjee, Thomas Lew, Rachel Luo,
Spencer M. Richards, Yixiao Sun, Edward Schmerling, and Marco Pavone

Stanford University
{rhnsinha,apoorva,somrita,thomas.lew,rsluo,
spenrich,alvinsun,schmrlng,pavone}@stanford.edu

Abstract. When testing conditions differ from those represented in training data, so-called out-of-distribution (OOD) inputs can mar the reliability of "black-box" learned components in the modern robotic autonomy stack. Therefore, coping with OOD data is an important challenge on the path towards reliable learning-enabled open-world autonomy. In this paper, we aim to demystify the topic of OOD data and associated challenges in the context of data-driven robotic systems, drawing connections to emerging paradigms in the ML community that study the effect of OOD data on learned models in isolation. We argue that as roboticists, we should reason about the overall *system-level* competence of a robot as it performs tasks in OOD conditions. We highlight key research questions around this system-level view of OOD problems to guide future research toward safe and reliable learning-enabled autonomy.

Keywords: out-of-distribution, learning-enabled robotics

1 Introduction

Machine learning (ML) systems are poised for widespread usage in the robot autonomy stack in the near future, driven by the successes of modern (deep) learning. For instance, decision-making algorithms in autonomous vehicles depend on ML-based perception and prediction models to estimate and forecast the state of the environment. As we as roboticists increasingly rely on ML models to contend with the unstructured and unpredictable real world, it is paramount that we also acknowledge the shortcomings of our models, especially when we hope to deploy robots alongside humans in safety-critical settings.

ML models can behave unreliably on data that is dissimilar from the training data — inputs termed *out-of-distribution* (OOD). This poses a significant challenge to deploying robots in the open world, e.g., as autonomous vehicles or home assistance robots, as such robots must interact with complex environments in conditions we cannot control or foresee. Coping with OOD inputs remains a key unsolved challenge on the critical path to reliable and safe open-world autonomy.

In this paper, we concretize the often nebulous notion of the OOD problem in robotics, drawing connections to existing approaches in the ML community. Critically, we advocate for a *system-level* perspective of OOD in robotics, which

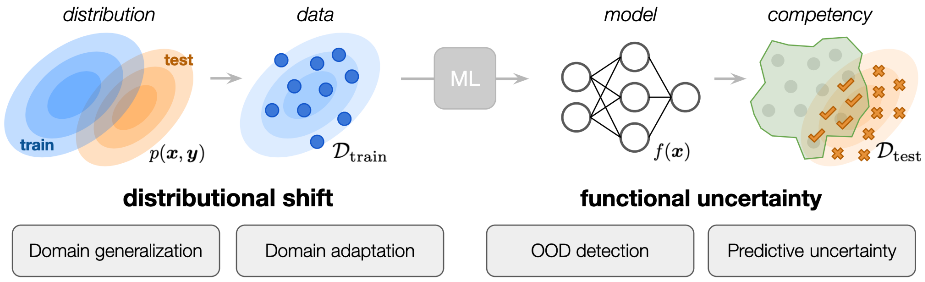

Fig. 1. Learning a predictive model f from a finite dataset poses challenges, especially in the presence of distributional shift. To address this, methods in the ML community consider both training and adapting models in anticipation of and response to distributional shift, or by considering methods that quantify functional uncertainty, by predicting when inputs are anomalous or quantifying uncertainty in the model's predictions.

considers the impacts of OOD data on downstream decision making and leverages the full autonomy stack to mitigate negative consequences. To this end, we present robotics research challenges at three timescales crucial to reliable open-world autonomy: (i) real-time decision-making, (ii) episodic interaction with an environment, and (iii) the data lifecycle as learning-enabled robots are deployed, evaluated, and retrained.

2 What Makes Data Out-Of-Distribution?

Well-engineered ML pipelines can produce models that generalize well to test data sampled i.i.d. from the same distribution as the training data. Consequently, when models *fail* to generalize at test time, we often attribute this to "OOD data" in a catch-all manner. What makes data OOD, and what causes these failures? In this section, we illustrate two concepts that structure our perspective on these questions using the notation of a standard supervised learning pipeline. Assume access to independent samples $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$ drawn from an underlying joint distribution P_{train} with density $p_{\text{train}}(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x} \in X$ and $\boldsymbol{y} \in Y$. In supervised learning, we fit a model $f : X \to Y$ on $\mathcal{D}_{\text{train}}$ and evaluate its performance on a test data set $\mathcal{D}_{\text{test}}$ drawn from P_{test} with density $p_{\text{test}}(\boldsymbol{x}, \boldsymbol{y})$.

Distributional Shifts: A *distributional shift* occurs when test data $\mathcal{D}_{\text{test}}$ is sampled from a distribution P_{test} that differs from P_{train}, thereby making $\mathcal{D}_{\text{test}}$ OOD and $\mathcal{D}_{\text{train}}$ *in-distribution* (ID). Shifts can corrupt the performance of the learned model f since it may no longer capture the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$ in the test data. Distributional shifts can reflect fundamental changes in the underlying data generating process (often termed *concept shift*), like when we learn dynamics of a robot that change due to actuator degradation. Alternatively, shifts can be limited to part of the generative process. A *covariate shift* describes when $p(\boldsymbol{x})$ changes while $p(\boldsymbol{y} \mid \boldsymbol{x})$ remains constant [1]. For instance, we might train a vision model on images collected during the day and deploy the model at night. Similarly, *label shift* occurs when $p(\boldsymbol{y})$ changes and $p(\boldsymbol{x} \mid \boldsymbol{y})$ does not, for example when deploying a pre-trained pedestrian detector in a new country where there are overall more pedestrians [2]. The language of distributional shifts is particularly suited to quantifying how population level statistics, like the expected loss of a model, are affected by changing conditions.

Functional Uncertainty: Since we do not have access to P_{test} directly and must learn a model f from a finite set of samples $\mathcal{D}_{\text{train}}$, we cannot be certain that f will make good predictions at test time. This offers a complementary view on the OOD problem; instead of reasoning about distributional differences, we aim to characterize the domain of competence of a particular f, i.e., when and where we can have confidence in its individual predictions, and conversely, when we are uncertain in its predictions. We refer to this as the *functional uncertainty* perspective on the OOD problem. Causes of high functional uncertainty are not rooted only in distributional notions; even when $P_{\text{test}} = P_{\text{train}}$, the model f may make poor predictions on rare inputs which were not well represented in the finite $\mathcal{D}_{\text{train}}$. Instead, functional uncertainty may arise from *epistemic* uncertainty, i.e., when we are unaware of the input-output relations that our models do not capture in the test domain. Of course, distributional shifts can increase the likelihood of encountering test data outside the domain of competence of f. Importantly, functional uncertainty is linked to how f is used, as evaluating competence requires a measure of test-time performance. While this measure could be generic (e.g., KL divergence between $f(\boldsymbol{x})$ and $p_{\text{test}}(\boldsymbol{y} \mid \boldsymbol{x})$), it can also be tailored to downstream utility functions.

3 Recent Trends in OOD in Machine Learning

The OOD problem is an open challenge in the ML community. Indeed, state-of-the-art models have been shown to be extremely sensitive to subtle distributional shifts [3,4]. In this section, we discuss core formulations and techniques from the ML literature tackling the OOD challenge, summarized in Figure 1.

3.1 Coping with Distributional Shift

Standard ML techniques are built around the often unrealistic assumption that $P_{\text{test}} = P_{\text{train}}$. A major line of ML research aims to relax this assumption to develop learning algorithms that can cope with distributional shifts.

Domain generalization considers the capacity of a model trained only on data from P_{train} to generalize to an *unknown* test-time data distribution P_{test}. Thus, domain generalization amounts to coping with distributional shift between train and test time. One salient research direction aims to improve *distributional robustness*, optimizing the worst-case performance within an envelope of distributional shifts [5]. A complementary research direction targets the root cause of poor generalization under distributional shift from a *causal inference* perspective: Learned models often pick up spurious correlations in $\mathcal{D}_{\text{train}}$, rather than the invariant cause and effect relations that govern the underlying process [6,7].

Domain adaptation aims to develop algorithms that leverage both the training dataset and some unlabeled test inputs $\{\boldsymbol{x}_i\}_{i=1}^{M} \overset{\text{iid}}{\sim} P_{\text{test}}$ to optimize the learned model f on P_{test}. Adapting f to the test inputs is a meta-approach that often yields drastic performance improvements with simple algorithms. For example, for covariate shift problems, the most elementary approach is to apply importance reweighting techniques to yield unbiased estimates of the model's risk under P_{test} [1]. Classic results in domain adaptation theory state that performance degradation under distributional shift is linked to how well a classifier can distinguish data from the train and test domains [8]. These results motivate methods which learn feature representations such that train and test data look similar [9], or learn transformations between the train and test domains [10].

3.2 Assessing Functional Uncertainty

Domain adaptation and generalization focus on methods to select or improve the learned model f in anticipation of or response to a changed data distribution P_{test}. Orthogonally, we can also consider methods that aim to characterize the functional uncertainty of a *particular* model f trained on data $\mathcal{D}_{\text{train}}$.

Detecting anomalous inputs: A key source of functional uncertainty lies in inputs that are dissimilar to those seen in the training data. *Anomaly detection* considers the challenge of predicting if an individual input is dissimilar to $\mathcal{D}_{\text{train}}$ [11,12]. This problem is also often called *out-of-distribution detection*, but many approaches do not explicitly model the training distribution, so we use the more generic term *anomaly detection.* We can measure dissimilarity from the training data via a distance metric, or by evaluating likelihoods under a learned parametric model of $p_{\text{train}}(\boldsymbol{x})$. These strategies are often applied in a learned feature space instead of directly on the inputs because modeling distances and distributions can be difficult for high-dimensional inputs, such as images.

Predictive Uncertainty: An alternative approach is to design a model f that directly outputs a measure of confidence in its predictions. However, it is challenging to ensure that the confidence measures we use to assess functional uncertainty remain calibrated in OOD regimes. For example, the softmax output distribution of classification networks is often confidently wrong on OOD data [13]. Besides calibration algorithms and design choices that encourage high predictive uncertainty on anomalous inputs, *Bayesian ML* offers an appealing approach to assess functional uncertainty under covariate shifts. This is because Bayesian methods allow us to quantify *epistemic uncertainty* by incorporating subjective prior beliefs to yield a posterior distribution $p(f \mid \mathcal{D}_{\text{train}})$ [14].

3.3 Evaluation

Researchers have developed benchmark datasets that contain train/test splits curated for qualitative semantic differences for evaluating OOD performance (e.g., see [15,16,4]). For example, the training set could include daytime images, while the test set could include nighttime images. Such data sets provide an intuitive foothold to develop algorithms by isolating reliability problems rooted

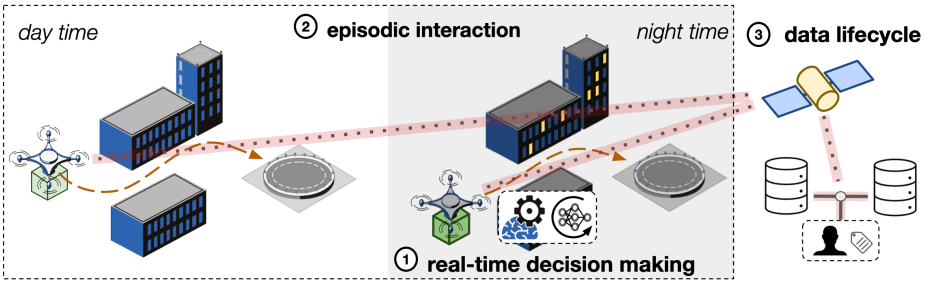

Fig. 2. Data-driven systems operating at different timescales. (1) Learning-enabled robotic systems must take actions and react to novel conditions in changing environments, requiring real-time OOD monitoring tools. (2) Long-horizon tasks (e.g., transporting a payload to a destination) require robotics OOD tools that consider episodic interactions, in which the typical assumption that inputs are drawn i.i.d. does not hold and time correlations should be accounted for. (3) Finally, learning-based models should be retrained to continuously improve the reliability of the overall robotic stack.

in OOD data. However, it is often unclear how methods tested on semantically OOD data will impact a robotic system downstream at deployment.

4 Open Challenges for OOD in Robotics

Robotics has always been centered on building *systems* that work well in the real world. Therefore, we argue for a *system-level* perspective on tackling OOD data in learning-enabled autonomy: Our ultimate goal is to reason about an ML-enabled autonomous system's reliability and competence as it operates in potentially shifted conditions. This perspective differs from the model-level paradigms in the ML community and thus presents unique challenges for the robotics community. We consider three different timescales at which data-driven robotic systems operate, as shown in Figure 2, each with distinct OOD challenges. We discuss each, drawing connections to methods from the ML community and highlighting key open research questions (**RQ**s) toward enabling autonomous systems to leverage ML while being robust to OOD conditions.

4.1 Real-time ML-Enabled Decision Making

Learning-enabled autonomous systems evaluate an ML model f in real-time on unseen inputs to make decisions. To maintain system-level competence at runtime, we need to reason about the downstream impact of individual inputs on the decision-making system. For example, failing to detect a pedestrian could be disastrous. Therefore, at the real-time timescale, we must reason about the competence of the full decision-making stack on individual inputs encountered at test time. This perspective is commonplace in robotics and suggests two key research questions centered around the functional uncertainty viewpoint on OOD.

RQ 1 (Averting OOD failures through Runtime Monitoring)**.** Can we leverage *full-stack* sensory information at runtime to detect if a decision system relying on a learned model f will perform poorly, before a failure occurs?

Without making restrictive assumptions on the environment, it is impossible to provably guarantee performance on unseen test-time conditions [17]. Therefore, a core aspect of achieving system-level competence involves developing systems to monitor the performance of a learning-enabled system at runtime. Assessing the functional uncertainty on the model's inputs is an important first step towards this goal, but is not sufficient to monitor the performance of the overall robotics stack. Instead, we need to reason about how functional uncertainty propagates through the decision-making system and devise goal-oriented measures of uncertainty on f that capture system-level performance. Indeed, the downstream impact of erroneous predictions may vary between systems or the current system state. Access to the full robotic autonomy stack also presents opportunities to use information from additional sensors besides the model's inputs to improve our assessment of functional uncertainty during operation.

RQ 2 (OOD Aware Decision Making)**.** Can we design decision-making systems compatible with runtime monitors robust to high functional uncertainty?

A robot must always choose an action to take, even if runtime monitors suggest that a learned component f is operating OOD. Thus, as roboticists, we must design systems where model uncertainties are assessed and accounted for during decision-making. This entails the joint design of real-time runtime monitors, uncertainty-aware decision-making algorithms, and fallback strategies. Since fallbacks may need to rely on redundancy or alternate sources of information, the problem of ensuring the safety and reliability of the aggregate autonomous system is a significantly more expansive challenge than that of characterizing the functional uncertainty of an ML model in isolation.

4.2 Episodic closed-loop interaction

Learning-enabled robots do not passively make predictions on a set of given individual inputs. Instead, they actively interact with their environment to perform tasks. Thus, reliable robotic systems should also reason about the influence of OOD conditions on the closed-loop decision-making system over extended periods of time. At this timescale, this *sequential decision-making* context induces key distinct research challenges for the robotics community.

RQ 3 (Temporally Correlated OOD events)**.** Can we develop methods that account for the temporal correlations between inputs when we repeatedly evaluate a learned model f under shifted conditions over the course of an episode?

As discussed in Section 3, considering population statistics like the expected loss under distributional shift is one of the core frameworks in ML research to study OOD performance. However, when a robot is deployed over an episode, the learned model's inputs will be correlated over time. Even in nominal conditions, these temporal correlations induce distributional shifts from training data. For example, while an ML perception model would likely be trained on a set of inputs from diverse weather conditions, an autonomous vehicle will likely only encounter one weather condition during a particular trip. Therefore, as roboticists, we should investigate how we can strengthen performance in anticipation of shifted conditions that affect the reliability of model outputs over the course of a trajectory, for example by assuring generalization across domains, or rapidly adapting to conditions faced at deployment.

RQ 4 (Mitigating Distributional Shifts)**.** Can we construct decision-making algorithms that mitigate distributional shifts between the training and deployment conditions to ensure the overall reliability of the deployed system?

Robotic systems often have agency to mitigate distributional shifts through decision-making. For example, a drone can avoid aggressive maneuvers in regions where it has limited data to mitigate the consequences of potential errors in a learned dynamics model f. By ensuring that learning-enabled components operate in distribution, the design of OOD monitors is simplified, the use of fallback strategies is reduced, and the reliability of the robotic stack is generally improved. To achieve this intelligent behavior, we must design methods to quantify and reason about the *domain of competency* of learned systems in a manner that is amenable to planning and decision making.

4.3 Data lifecycle

Finally, beyond interactions during individual episodes, we can consider long-term cycles over which data-driven robotic systems are deployed, evaluated, improved, and deployed again. In this context, we view the development process as a feedback loop, potentially with human experts in the loop. At this scale, our goal is to use data collected during operation to improve the system's overall performance across novel, rare, or shifted conditions.

RQ 5 (Leveraging Operational Data)**.** How can we use data collected during operation in diverse tasks and contexts to improve the robustness and quality of learned models?

Retraining components on new data collected during operation can mitigate the influence of OOD conditions by reducing functional uncertainty and ensuring that training data matches test conditions. However, simply appending operational data to $\mathcal{D}_{\text{train}}$ may not be enough to avoid learning spurious correlations or improve performance on extremely rare failure modes. Therefore, we should also aim to increase the diversity of the data and leverage the fact that data collected during different episodes of robot execution represents a set of diverse test-time contexts, which can be naturally grouped into different operational domains. This task-specific structure lends itself well to a variety of approaches to improve domain generalization, like multi-task, meta-, or causal learning, which can yield a model that is able to better generalize to new conditions.

RQ 6 (Efficient Data Collection)**.** How do we select what operational data we should use to efficiently improve our models?

Robotic fleets collect tremendous amounts of data during operation, not all of which can be stored or labeled to improve the performance of the autonomy stack. Moreover, collecting more data through robot deployments is costly and can see diminishing returns. Therefore, we need to understand how to efficiently collect data during operation and judiciously choose which data to flag for labeling. This problem has strong connections to research on estimating functional uncertainty in ML models, as the most informative inputs to label correspond to those on which the model f is most uncertain.

5 Conclusion

The recurring theme across these timescales is that the *full-stack* nature of robotics requires a *system-level* perspective on the OOD problem. We argue that roboticists should embrace this system-level perspective: Investigate both how OOD data impacts the reliability of the full autonomy stack and how to leverage the full autonomy stack to mitigate negative consequences. While these research questions are challenging and involve all aspects of the autonomy stack, they represent necessary steps towards a future where we can safely and reliably leverage ML to enable true open-world autonomy.

References

1. H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 2000.
2. M. Saerens et al. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 2002.
3. B. Recht et al. Do ImageNet classifiers generalize to ImageNet? In *ICML*, 2019.
4. J. Miller et al. Accuracy on the line: On the strong correlation between out-of-distribution and in-distribution generalization. In *ICML*, 2021.
5. A. Ben-Tal et al. Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 2013.
6. Judea Pearl. *Causality.* Cambridge University Press, 2009.
7. M. Arjovsky et al. Invariant risk minimization. *arXiv:1907.02893*, 2019.
8. S. Ben-David et al. *A theory of learning from different domains.* Machine Learning 79, 2010.
9. Y. Ganin et al. Domain-adversarial training of neural networks. *JMLR*, 2016.
10. J. Hoffman et al. CyCADA: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018.
11. M. Salehi et al. A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *arXiv:2110.14051*, 2021.
12. L. Ruff et al. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 2021.
13. Y Ovadia et al. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.
14. Moloud Abdar et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 2021.
15. D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.
16. P. Koh et al. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021.
17. S. Seshia et al. Towards verified artificial intelligence. *arXiv:1606.08514*, 2016.