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

Anonymous submission

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 robot autonomy stack. Therefore, coping with OOD data is an important challenge on the path towards trustworthy learning-enabled open-world autonomy. In this paper, we aim to demystify the topic of OOD data and its 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.

1 Introduction

Machine learning (ML) systems are poised for widespread usage in robot autonomy stacks in the near future, driven by the successes of modern deep learning. For instance, decision-making algorithms in autonomous vehicles rely on ML-based perception and prediction models to estimate and forecast the state of the environment. As we increasingly rely on ML models to contend with the unstructured and unpredictable real world in robotics, 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.

In particular, ML models may behave unreliably on data that is dissimilar from the training data — inputs commonly 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 and largely unsolved challenge on the critical path to reliable and safe open-world autonomy. However, there is no generally-agreed-upon precise definition of what makes data OOD; instead, its definition is often left implicit and varies between problem formalisms and application contexts.

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 data in robotics, which considers the impacts of OOD data on downstream decision making and allows for leveraging components throughout the full autonomy stack to mitigate negative consequences. To this end, we present robotics research challenges at three timescales crucial to deploying 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.

We emphasize that this paper does not represent a comprehensive survey of existing paradigms and literature on OOD topics in machine learning or robotics; in fact, many of the OOD topics that we discuss, like runtime-monitoring of perception systems (Rahman, Corke, and Dayoub 2021) or heuristic uncertainty quantification of deep neural networks (Abdar et al. 2021), constitute exhaustively surveyed subfields in their own right. This work differs in that we provide an overview of the core considerations and system-wide challenges that we see as essential areas of robotics research activity for the coming years, rather than survey specific styles of analysis or approaches tailored towards particular submodules in the autonomy stack.

2 Running Examples

To better describe the challenges that OOD data creates in learning-enabled robotic systems, we use the two future autonomy systems shown in Figure 1 as running examples in this paper. These conceptual examples highlight the plurality of applications and design paradigms used to leverage ML in the design of robotic systems.

Autonomous Drone Delivery Service: Firstly, we consider an autonomous drone delivering packages in a city. As illustrated in Figure 1, this robot uses several learning-enabled components in its autonomy stack. The delivery drone has to make explainable decisions and meet stringent safety requirements by regulatory agencies to be deployed among humans. Crucially, to maintain these reliability requirements in rare and unforeseen circumstances, the drone needs mechanisms to detect and manage OOD inputs.

Robotic Manipulators Assisting in the Home: Secondly, we consider the deployment of robotic manipulators to assist with various tasks in and around the home, as shown

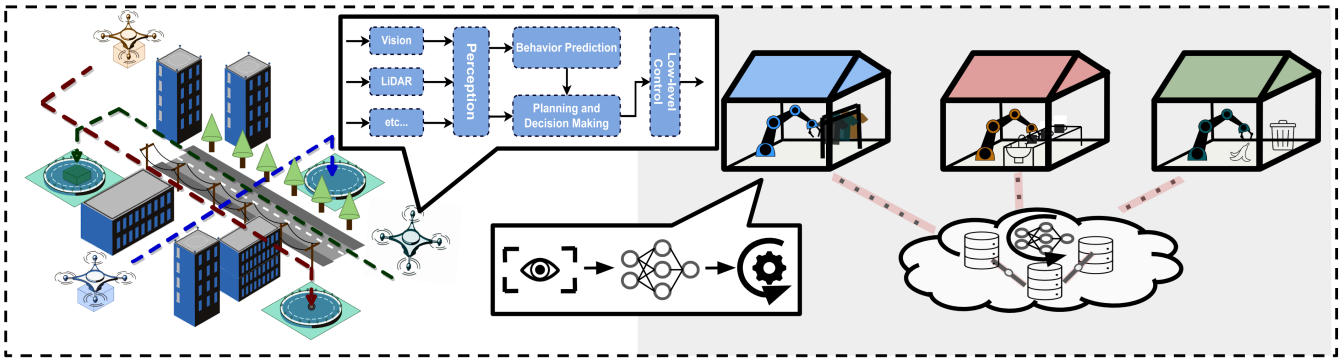

Figure 1: **Left:** A future drone delivery system. It uses a modular autonomy stack consisting of 1) a perception system that builds an understanding of the drone's state and its environment, 2) a prediction module that makes inferences about the behavior of other agents and objects in the drone's environment, 3) a high-level planning and decision-making module, and 4) low-level controllers that actuate the drone's propellers to control the drone's trajectory. At deployment, the drone must safely navigate many obstacles such as other delivery drones, power lines, and trees, not all of which can be anticipated at design time. **Right:** Robotic manipulators assisting in the home. The manipulators are controlled end-to-end by directly passing the observations of the robot through a policy network that outputs the actions the robot should take. At deployment, the manipulators perform a wide range of household tasks like folding and hanging clothes, washing dishes, and cleaning up trash. Therefore, engineers train the manipulator policy by creating targeted experiments to collect a large and diverse training dataset of many objects to manipulate and tasks to complete.

in Figure 1. The manipulators' tasks are so diverse and unstructured that we consider a general manipulation policy trained in an end-to-end fashion in a controlled environment, as commonly considered in the reinforcement learning (RL) community. When we deploy these manipulators in people's homes, the environments and contexts that these robots encounter invariably differ from the lab or simulated environments used for training, which can markedly impact the system's performance. Therefore, ensuring reliable performance in OOD test environments is a crucial aspect of the design challenge.

3 What Makes Data Out-Of-Distribution?

Well-engineered ML pipelines 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*). Concepts can shift discontinuously, like when important unobserved features change between train and test, or they can slowly drift over time. For example, when we use a model-based approach for lower-level control of the delivery drone, slowly degrading actuators can make the predictions of a learned dynamics model dangerously inaccurate. 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 (Shimodaira 2000). For instance, we might train a vision model for the delivery drone on images collected during the day but deploy the delivery drone using 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 (Saerens, Latinne, and Decaestecker 2002). 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

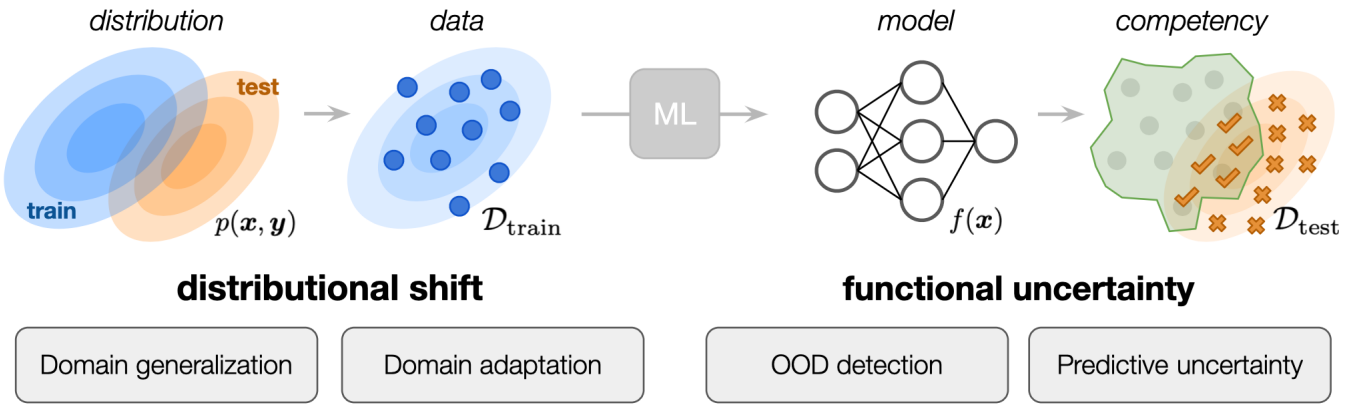

Figure 2: Learning a predictive model f from a finite dataset poses challenges, especially in the presence of distributional shift. To address this, methods in the ML community consider both training and adapting models in anticipation of and response to distributional shift. Besides improving models on OOD data, other approaches consider methods that quantify functional uncertainty by predicting when inputs are anomalous or quantifying uncertainty in the model's predictions.

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 can 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.

4 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 (e.g., see (Torralba and Efros 2011; Geirhos et al. 2020; Hendrycks and Dietterich 2019; Recht et al. 2019; Miller et al. 2021) and the references therein). In this section, we discuss classical and core formulations and techniques from the ML literature that guide the ML community's approach to tackling the OOD challenge, summarized in Figure 2.

4.1 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. To make this problem approachable, we need to make assumptions on how much P_{test} can reasonably differ from P_{train}. For example, one salient research direction aims to improve *distributional robustness*, optimizing the worst-case performance within an envelope of distributional shifts to guarantee OOD performance (Ben-Tal et al. 2013; Duchi and Namkoong 2021). However, it is often unclear how large distributional uncertainty sets should be when conditions shift, a topic that roboticists working on applications should address. To circumvent such ambiguity, it is common to consider the robustness behavior on subpopulations of the training data instead (Sagawa* et al. 2020). 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 (Pearl 2009; Peters, Bühlmann, and Meinshausen 2016; Arjovsky et al. 2020). For example, the vision-based robotic manipulator could rely on features in the background of the image to identify object types in the foreground (e.g, cooking items usually appear in kitchens), and thus fail to generalize to reasonable distribution shifts where the background changes (e.g., a pan on a sofa). Empirically, domain generalization often improves most when we (pre)train larger models on larger, more diverse datasets and infuse domain knowledge by encoding invariances explicitly or via data augmentations and self-supervised pretraining tasks (Miller et al. 2021; Zhou et al. 2022; Gulrajani and Lopez-Paz 2021).

Domain adaptation aims to develop algorithms that leverage both the training dataset and some (usually 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}. Domain adaptation therefore typically requires a priori availability of some test domain data. This can occur when we make batched predictions on a given test set, or, for example, when we deploy the drone delivery service in a new country and have a small budget for running pre-deployment trials. Adapting f to the test inputs is a paradigm that often yields drastic performance improvements with simple algorithms because the test domain data allows us to make inferences about a model's performance on P_{test}. 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}

(Shimodaira 2000). 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 (Ben-David et al. 2010; Redko et al. 2020). These results motivate methods which learn feature representations such that train and test data look similar (Ganin et al. 2016), or learn transformations between the train and test domains (Hoffman et al. 2018). More broadly, progress on algorithms that adapt models in response to shifted conditions is not limited to the batched setting (surveyed extensively in (Wilson and Cook 2020)). In particular, *continual or lifelong learning* algorithms seek to adapt f over time in response to evolving distributions (De Lange et al. 2022; Lesort et al. 2020), and *meta-learning* considers the design of algorithms that can rapidly adapt models to new distributions given separate datasets from related domains (Finn, Abbeel, and Levine 2017).

4.2 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 $\mathcal{D}_{\text{train}}$ (Salehi et al. 2021; Ruff et al. 2021; Yang et al. 2021). 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 in various ways, such as 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. To ensure confidence scores are correct, some approaches ensure a model's predictions are *calibrated*, i.e. that the predictive uncertainty matches the model's error rate (Guo et al. 2017). Others, like conformal inference, compute prediction intervals containing the correct label with high probability (Balasubramanian, Ho, and Vovk 2014). However, it is generally 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 (Ovadia et al. 2019). Therefore, many empirical studies investigate design choices that encourage high predicted uncertainty on inputs that are dissimilar to training inputs, such as specific model architectures, auxiliary losses, and regularizers (Abdar et al. 2021). 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 $p(f \mid \mathcal{D}_{\text{train}})$ (Abdar et al. 2021). However, scaling Bayesian methods to large models is a computationally challenging task. Therefore, many methods approximate the Bayesian posterior, for example, through ensembling, or monte-carlo dropout (Lakshminarayanan, Pritzel, and Blundell 2017; Gal and Ghahramani 2016).

4.3 Evaluation

Researchers have developed benchmark datasets that contain train/test splits curated for qualitative semantic differences for evaluating OOD performance (e.g., see (Hendrycks and Dietterich 2019; Koh et al. 2021; Miller et al. 2021)). OOD test sets can include synthetic corruptions like motion blur, Gaussian noise, and other perturbations. In addition, many datasets may test robustness to naturally occurring distribution shifts, like when we train the delivery drone's bird detection model only on images of land birds and include images of waterbirds in the test set. Such data sets provide an intuitive foothold to develop algorithms by isolating reliability problems rooted in OOD data. However, it is often unclear how methods tested on semantically OOD data will impact a robotic system downstream at deployment.

5 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 when it applies learned models in a feedback loop over time, as it operates in potentially shifted conditions. This perspective differs from the model-level paradigms in the ML community aimed at quantifying how a model's accuracy degrades on independently-sampled OOD data because learned models only constitute individual components of a complex autonomy stack. Therefore, system-level perspectives present unique challenges for the robotics community related to detecting OOD conditions, responding to them to avert system failures, and improving the robotic system's OOD closed-loop performance as a whole. We illustrate these challenges by considering three different timescales at which data-driven robotic systems operate, as shown in Figure 3, each with distinct OOD challenges for robotics. We discuss each timescale, drawing connections to methods from the ML community and highlighting key open research questions (**RQ** s) toward autonomous systems that leverage ML while being robust to the OOD conditions they will inevitably encounter. In addition, we examine various aspects of the RQs using our running examples and briefly touch upon recent research trends to contextualize the RQs. We emphasize that this discussion is not an exhaustive survey of existing literature but rather a brief discussion to underscore the significance of the RQs.

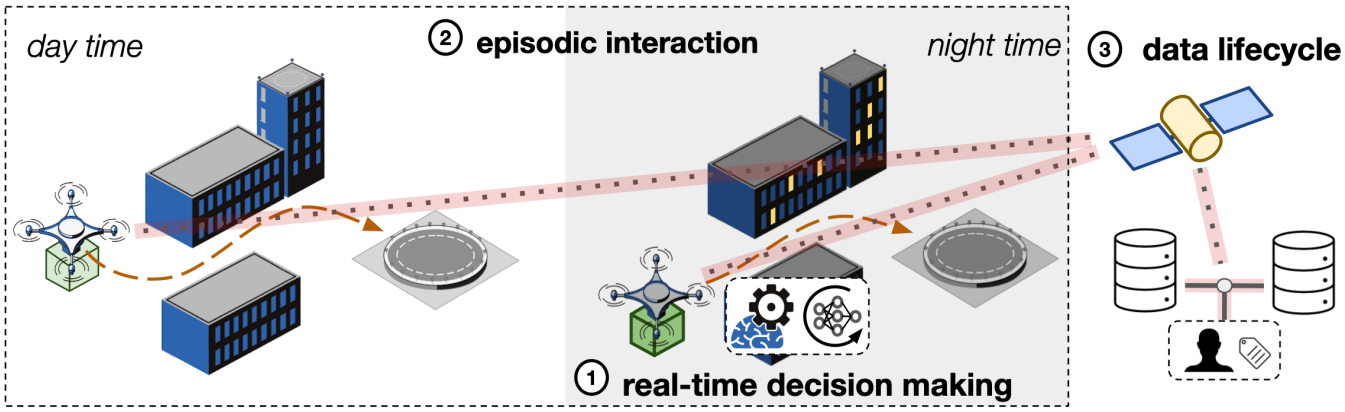

Figure 3: Data-driven systems operating at different timescales. (1) Learning-enabled robotic systems must take actions and react to novel conditions in changing environments, requiring real-time OOD monitoring tools. (2) Long-horizon tasks (e.g., transporting a payload to a destination) require robotics 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 offline to continuously improve the reliability of the overall robotic stack.

5.1 Real-time ML-Enabled Decision Making

To maintain system-level competence at runtime, we need to reason about the downstream impact of individual OOD inputs on the decision-making system in real-time. For example, a failure of the delivery drone to detect a pedestrian could be disastrous; we need to construct safeguards that ensure inference errors do not lead to system failure. Therefore, at the real-time timescale, we need to monitor the competence of the full decision-making stack on individual inputs encountered at test time. Even though runtime monitoring is commonplace in robotics, its application to mitigate the effect of OOD data on state-of-the-art learned components 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?

Because we generally cannot identify all aspects of the robot's environment that affect the reliability of its ML components, provable safety guarantees are virtually unattainable for autonomous systems without making restrictive assumptions (Seshia, Sadigh, and Sastry 2020). Indeed, the failures caused by conditions that were not represented at design time –be it in training data or in simulated test scenarios– are generally what we attribute as OOD. Therefore, we view algorithms for monitoring autonomous systems at runtime as a core aspect of maintaining system-level competence in OOD regimes.

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.

Additional Examples: Consider the scenario of the delivery drone's perception network failing to detect OOD object types not seen at training time. At high speeds, missing a detection on a single image can make obstacle avoidance impossible. Runtime monitors as suggested in **RQ 1** that flag when the functional uncertainty of the visual object detector on a specific input is high, signaling that the model outputs are unreliable, can be critical to avoid catastrophic failures. However, not every missed detection will affect the same consequences: a missed detection of an OOD bird breed far away is less likely to cause a collision than a missed detection of a nearby tree (complicating **RQ 1**). When the runtime monitor signals that the object detection system is dangerously inaccurate, the drone should land or continue safe operation in a degraded state. Certifying that a fallback strategy does not cause additional hazards, such as ensuring the drone does not land on busy roads with limited sensing, re-

quires the system-level considerations outlined in **RQ 2**.

Knowing that particular objects are OOD for the robot manipulator can help it decide which objects it can reliably manipulate. This knowledge can allow the robot to abstain from handling OOD objects instead of dropping and damaging them. In addition, we can build system-level checks around the manipulator policy to sanity-check its decisions. For example, we could compare to grasps computed using more classical techniques or leverage additional sensing modalities to estimate when the robot can or cannot successfully manipulate an object.

Recent Trends: As discussed in Section 4, quantifying functional uncertainty of a model on OOD inputs is a lively field of study, including anomaly detection (as surveyed in (Salehi et al. 2021; Ruff et al. 2021; Hodge and Austin 2004)) and heuristics for predictive uncertainty like approximate Bayesian inference (e.g., (Sharma, Azizan, and Pavone 2021; Lakshminarayanan, Pritzel, and Blundell 2017; Amini et al. 2020)) to name a few. However, such methods generally only apply to covariate shifts. In addition, robotics-focused monitoring methods that leverage additional sensors to learn how models are innacurate or check consistency among modules (as surveyed in (Rahman, Corke, and Dayoub 2021)), or learn to predict or recognize system failures also show promise (Luo et al. 2021; Farid et al. 2022). However, these early approaches are often not goal-oriented, heuristic, or unverifiable under distribution shift. Moreover, while many existing control theoretic approaches provide safety filters that interfere with black-box learned policies to correct trajectories in settings where system dynamics and state are known (e.g., (Leung et al. 2020; Fisac et al. 2019), certifiably closing the loop on runtime monitors and fallbacks in complex systems like our drone delivery example is a broadly understudied problem.

5.2 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 4, 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 we deploy a robot over an episode, the learned model's inputs will be correlated over time, violating the standard ML assumption that test samples are i.i.d. 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 shuffled inputs from diverse weather conditions from many trips, 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. In addition, we consider developing methods that certifiably detect performance-impacting shifts during execution without i.i.d. assumptions as a largely open problem.

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.

Additional Examples: Externally shifting conditions, such as those that occur when we train the drone's vision system on daytime images and deploy at night, will consistently degrade the perception system. **RQ 3** asks how we can distinguish consistent model errors induced by OOD conditions from sporadic errors, which may be tolerable. How many contiguous missed detections will induce a failure, and how do we anticipate this before it is too late? In addition, distributional shifts can stem from the fact that the drone needs to use some policy for test deployments to collect data: The delivery drone might use a slow and conservative policy to collect the data to learn interaction models for other agents. If the drone flies very aggressively using these models, the closed-loop trajectory distribution will shift, making the interaction models dangerously inaccurate. Exploiting the drone's ability to control this shift is **RQ 4**'s focus.

For the robotic manipulator, every household deployment represents shifted conditions from the environment in which we developed the policy: clothing styles and typical pots and pans vary between households and countries. In the context of **RQ 3** and **RQ 4**, we should study how we can quickly adapt the robot to the shifted or evolving conditions and expand operations to new task contexts reliably.

Recent Trends: Uncertainty in a robot's dynamics model or surroundings are forms of temporally correlated OOD uncertainties eminent both in the learning-based control and RL communities through topics like dynamics learning and sim2real transfer (Brunke et al. 2022). However, quantifying and managing the system-level effects of temporally correlated OOD data in complex sensing modalities like perception systems is an expansive open problem, with early steps including contributions like (Farid, Veer, and Majum-

dar 2022; Podkopaev and Ramdas 2022) aimed at detecting shifted distributions between task executions rather than during long-term deployments. In addition, the distributional shift induced by a change in data collection and test policies is a core framework through which many robot learning problems, like imitation learning (Ross, Gordon, and Bagnell 2011) and offline RL (Levine et al. 2020), are studied. More broadly, ego-influenced distributional shifts may affect any learned model in an autonomy stack, not just systems trained end-to-end.

5.3 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 distributionally robust, 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 or testing 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.

Additional Examples: Daily varying wind and weather conditions can significantly affect the dynamics of a delivery drone carrying a large payload in an a priori hard-to-model fashion. In line with **RQ 5**, we can use the episodically collected trajectory data more effectively by using meta-learning techniques to learn structure in the weather disturbance dynamics so we can adapt online more rapidly, improving control performance. However, a small delivery drone has severely limited data storage capacity. Therefore, **RQ 6** concerns how we should select which data to store to improve the system and what data we discard in real-time. Furthermore, it is insufficient to keep a buffer and upload it when an incident or failure occurs because reliability requirements make system failures extremely rare, even when learning-enabled subcomponents often make errors.

Alternatively, consider an example where we train the robot manipulator to complete ten separate household tasks. Suppose we naively train to maximize the average performance across tasks. In that case, we might greedily sacrifice performance on one task, like sweeping the floor, because improvements on another task, like loading dishwashers, outweigh the cost. Then, during deployment, some users might request the robot to sweep the floor much more than they use it to load dishes. The shifted task distribution these users request will result in poor overall performance, even though the engineers in the lab see a high average performance. Instead, as is the focus in multi-task learning or subgroup distributional robustness, the robot should be trained in line with **RQ 5** so that performance is consistently good across tasks. Then, users will always observe good performance no matter their task proclivities. In addition, engineers will operate on a fixed budget for experimentation and training, so they must judiciously design experiments that maximize the robot's performance, a facet of **RQ 6**.

Recent Trends: Roboticists have already started leveraging some of the ML community trends aimed at improving generalization when distributions vary across deployments beyond augmenting datasets with new operational data and retraining: Some approaches learn consistent patterns by considering separate losses for each trajectory, for example, by applying meta-learning to more rapidly learn dynamics models (Richards et al. 2021) or policies (Nagabandi et al. 2019) across environments, leveraging causal inference techniques to identify generalizable state and task representations (as surveyed in (Stocking, Gopnik, and Tomlin 2022), (Kirk et al. 2021)), or through multi-task learning (Rusu et al. 2016; Ahn et al. 2022). Finally, active learning techniques (Settles 2012) need to be tailored to robotics to collect and label data efficiently. Ongoing efforts in these areas underscore the significance of the data lifecycle challenges in robotics and make progress on specific components in the autonomy stack.

6 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

Abdar, M.; Pourpanah, F.; Hussain, S.; Rezazadegan, D.; Liu, L.; Ghavamzadeh, M.; Fieguth, P.; Cao, X.; Khosravi, A.; Acharya, U. R.; Makarenkov, V.; and Nahavandi, S. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76: 243–297.

Ahn, M.; et al. 2022. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. arXiv:2204.01691.

Amini, A.; Schwarting, W.; Soleimany, A.; and Rus, D. 2020. Deep Evidential Regression. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 14927–14937. Curran Associates, Inc.

Arjovsky, M.; Bottou, L.; Gulrajani, I.; and Lopez-Paz, D. 2020. Invariant Risk Minimization. arXiv:1907.02893.

Balasubramanian, V. N.; Ho, S.-S.; and Vovk, V. 2014. *Conformal Prediction for Reliable Machine Learning*. Morgan Kaufmann.

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Wortman Vaughan, J. 2010. *A theory of learning from different domains*. Machine Learning 79.

Ben-Tal, A.; den Hertog, D.; Waegenaere, A. D.; Melenberg, B.; and Rennen, G. 2013. Robust Solutions of Optimization Problems Affected by Uncertain Probabilities. *Management Science*, 59(2): 341–357.

Brunke, L.; Greeff, M.; Hall, A. W.; Yuan, Z.; Zhou, S.; Panerati, J.; and Schoellig, A. P. 2022. Safe Learning in Robotics: From Learning-Based Control to Safe Reinforcement Learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1): null.

De Lange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2022. A Continual Learning Survey: Defying Forgetting in Classification Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7): 3366–3385.

Duchi, J. C.; and Namkoong, H. 2021. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3): 1378 – 1406.

Farid, A.; Snyder, D.; Ren, A. Z.; and Majumdar, A. 2022. Failure Prediction with Statistical Guarantees for Vision-Based Robot Control.

Farid, A.; Veer, S.; and Majumdar, A. 2022. Task-Driven Out-of-Distribution Detection with Statistical Guarantees for Robot Learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, 970–980. PMLR.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1126–1135. PMLR.

Fisac, J. F.; Akametalu, A. K.; Zeilinger, M. N.; Kaynama, S.; Gillula, J.; and Tomlin, C. J. 2019. A General Safety Framework for Learning-Based Control in Uncertain Robotic Systems. *IEEE Transactions on Automatic Control*, 64(7): 2737–2752.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Balcan, M. F.; and Weinberger, K. Q., eds., *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, 1050–1059. New York, New York, USA: PMLR.

Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; and Lempitsky, V. 2016. Domain-Adversarial Training of Neural Networks. *J. Mach. Learn. Res.*, 17(1): 2096–2030.

Geirhos, R.; Jacobsen, J.-H.; Michaelis, C.; Zemel, R.; Brendel, W.; Bethge, M.; and Wichmann, F. A. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11): 665–673.

Gulrajani, I.; and Lopez-Paz, D. 2021. In Search of Lost Domain Generalization. In *International Conference on Learning Representations*.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, 1321–1330. JMLR.org.

Hendrycks, D.; and Dietterich, T. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. *Proceedings of the International Conference on Learning Representations*.

Hodge, V.; and Austin, J. 2004. A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, 22(2): 85–126.

Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A.; and Darrell, T. 2018. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 1989–1998. PMLR.

Kirk, R.; Zhang, A.; Grefenstette, E.; and Rocktäschel, T. 2021. A Survey of Generalisation in Deep Reinforcement Learning. arXiv:2111.09794.

Koh, P. W.; et al. 2021. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 5637–5664. PMLR.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Lesort, T.; Lomonaco, V.; Stoian, A.; Maltoni, D.; Filliat, D.; and Díaz-Rodríguez, N. 2020. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information Fusion*, 58: 52–68.

Leung, K.; Schmerling, E.; Zhang, M.; Chen, M.; Talbot, J.; Gerdes, J. C.; and Pavone, M. 2020. On Infusing Reachability-Based Safety Assurance within Planning Frameworks for Human-Robot Vehicle Interactions. *Int. Journal of Robotics Research*, 39: 1326–1345.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. arXiv:2005.01643.

Luo, R.; Zhao, S.; Kuck, J.; Ivanovic, B.; Savarese, S.; Schmerling, E.; and Pavone, M. 2021. Sample-Efficient Safety Assurances using Conformal Prediction. arXiv:2109.14082.

Miller, J. P.; Taori, R.; Raghunathan, A.; Sagawa, S.; Koh, P. W.; Shankar, V.; Liang, P.; Carmon, Y.; and Schmidt, L. 2021. Accuracy on the Line: on the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 7721–7735. PMLR.

Nagabandi, A.; Clavera, I.; Liu, S.; Fearing, R. S.; Abbeel, P.; Levine, S.; and Finn, C. 2019. Learning to Adapt in Dynamic, Real-World Environments through Meta-Reinforcement Learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Ovadia, Y.; Fertig, E.; Ren, J.; Nado, Z.; Sculley, D.; Nowozin, S.; Dillon, J. V.; Lakshminarayanan, B.; and Snoek, J. 2019. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty under Dataset Shift. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc.

Pearl, J. 2009. *Causality*. Cambridge University Press.

Peters, J.; Bühlmann, P.; and Meinshausen, N. 2016. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5): 947–1012.

Podkopaev, A.; and Ramdas, A. 2022. Tracking the risk of a deployed model and detecting harmful distribution shifts. In *International Conference on Learning Representations*.

Rahman, Q. M.; Corke, P.; and Dayoub, F. 2021. Run-Time Monitoring of Machine Learning for Robotic Perception: A Survey of Emerging Trends. *IEEE Access*, 9: 20067–20075.

Recht, B.; Roelofs, R.; Schmidt, L.; and Shankar, V. 2019. Do ImageNet Classifiers Generalize to ImageNet? In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 5389–5400. PMLR.

Redko, I.; Morvant, E.; Habrard, A.; Sebban, M.; and Bennani, Y. 2020. A survey on domain adaptation theory. *CoRR*, abs/2004.11829.

Richards, S. M.; Azizan, N.; Slotine, J.-J. E.; and Pavone, M. 2021. Adaptive-Control-Oriented Meta-Learning for Nonlinear Systems. In *Robotics: Science and Systems*. Virtual.

Ross, S.; Gordon, G.; and Bagnell, D. 2011. 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*, volume 15 of *Proceedings of Machine Learning Research*, 627–635. Fort Lauderdale, FL, USA: PMLR.

Ruff, L.; Kauffmann, J. R.; Vandermeulen, R. A.; Montavon, G.; Samek, W.; Kloft, M.; Dietterich, T. G.; and Müller, K.-R. 2021. A Unifying Review of Deep and Shallow Anomaly Detection. *Proceedings of the IEEE*, 109(5): 756–795.

Rusu, A. A.; Colmenarejo, S. G.; Gülçehre, Ç.; Desjardins, G.; Kirkpatrick, J.; Pascanu, R.; Mnih, V.; Kavukcuoglu, K.; and Hadsell, R. 2016. Policy Distillation. In Bengio, Y.; and LeCun, Y., eds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Saerens, M.; Latinne, P.; and Decaestecker, C. 2002. Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure. *Neural Computation*, 14: 21–41.

Sagawa*, S.; Koh*, P. W.; Hashimoto, T. B.; and Liang, P. 2020. Distributionally Robust Neural Networks. In *International Conference on Learning Representations*.

Salehi, M.; Mirzaei, H.; Hendrycks, D.; Li, Y.; Rohban, M. H.; and Sabokrou, M. 2021. A Unified Survey on Anomaly, Novelty, Open-Set, and Out-of-Distribution Detection: Solutions and Future Challenges. arXiv:2110.14051.

Seshia, S. A.; Sadigh, D.; and Sastry, S. S. 2020. Towards Verified Artificial Intelligence. arXiv:1606.08514.

Settles, B. 2012. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1): 1–114.

Sharma, A.; Azizan, N.; and Pavone, M. 2021. Sketching curvature for efficient out-of-distribution detection for deep neural networks. In de Campos, C.; and Maathuis, M. H., eds., *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, 1958–1967. PMLR.

Shimodaira, H. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90: 227–244.

Stocking, K. C.; Gopnik, A.; and Tomlin, C. 2022. From Robot Learning To Robot Understanding: Leveraging Causal Graphical Models For Robotics. In Faust, A.; Hsu, D.; and Neumann, G., eds., *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, 1776–1781. PMLR.

Torralba, A.; and Efros, A. A. 2011. Unbiased look at dataset bias. In *CVPR 2011*, 1521–1528.

Wilson, G.; and Cook, D. J. 2020. A Survey of Unsupervised Deep Domain Adaptation. *ACM Trans. Intell. Syst. Technol.*, 11(5).

Yang, J.; Zhou, K.; Li, Y.; and Liu, Z. 2021. Generalized Out-of-Distribution Detection: A Survey. *arXiv preprint arXiv:2110.11334*.

Zhou, K.; Liu, Z.; Qiao, Y.; Xiang, T.; and Loy, C. C. 2022. Domain Generalization: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–20.