Towards a theory of out-of-distribution learning

Ali Geisa[†,1], Ronak Mehta[†,1], Hayden S. Helm[1], Jayanta Dey[1], Eric Eaton[2], Jeffery Dick[3], Carey E. Priebe[1], Joshua T. Vogelstein[1,*]

Abstract. What is learning? 20^{th} century formalizations of learning theory—which precipitated revolutions in artificial intelligence—focus primarily on *in-distribution* learning, that is, learning under the assumption that the training data are sampled from the same distribution as the evaluation distribution. This assumption renders these theories inadequate for characterizing 21^{st} century real world data problems, which are typically characterized by evaluation distributions that differ from the training data distributions (referred to as out-of-distribution learning). We therefore make a small change to existing formal definitions of learnability by relaxing that assumption. We then introduce **learning efficiency** (LE) to quantify the amount a learner is able to leverage data for a given problem, regardless of whether it is an in- or out-of-distribution problem. We then define and prove the relationship between generalized notions of learnability, and show how this framework is sufficiently general to characterize transfer, multitask, meta, continual, and lifelong learning. We hope this unification helps bridge the gap between empirical practice and theoretical guidance in real world problems. Finally, because biological learning continues to outperform machine learning algorithms on certain OOD challenges, we discuss the limitations of this framework vis-á-vis its ability to formalize biological learning, suggesting multiple avenues for future research.

1 Introduction The study of machine learning (ML) has enabled remarkable progress in artificial intelligence (AI), including revolutions in image recognition [1], natural language processing [2], medical diagnostics [3], autonomous control [4], and protein folding [5]. But empirical progress often outpaces theoretical understanding. Indeed, much of the progress of the last decade in ML/AI remains to be explained [6]. Moreover, the vast majority of the advances in ML/AI focus on in-distribution learning, that is, learning where the training and test data are assumed to be sampled from the same distribution.

However, in the real world, such in-distribution learning problems are quite narrow in scope; we would prefer to have ML/AI solutions that can solve out-of-distribution (OOD) learning problems [7]. Colloquially, OOD learning operates in a regime in which it assumes that training data are sampled from a distribution that differs from the evaluation data distribution. Current deep learning systems still struggle to adapt and generalize to seemingly trivial distributional changes [8, 9]. Moreover, as Thrun lamented in his seminal paper in 1996 introducing lifelong learning [10], the existing learning theory was inadequate to characterize this kind of learning. Since then, there have been a number of efforts to formalize various OOD learning scenarios, most notably Baxter [11], and quite recently Arjovsky [12]. However, most theoretical work on OOD learning focuses on a specific special case (such as meta or invariant learning). The lack of a unifying framework characterizing each of these different OOD learning scenarios has contributed to a number of ongoing challenges.

First, often the precise goals of various algorithms are unclear. For in-distribution learning, the goal is always clear: minimize generalization error. Out-of-distribution learning scenarios, however, are more complex. For example, when comparing two different OOD learning algorithms, if they both observe some in-distribution data, then one could out perform the other in terms of out-of-distribution generalization error by one of several possible mechanisms: (i) it could have better priors and/or inductive biases, (ii) it could leverage the in-distribution data more efficiently, or (iii) it could leverage the out-of-distribution data more efficiently. Simply comparing accuracy fails to quantify the extent to which a given algorithm is

[†] denotes equal contribution, [*] corresponding author: jovo@jhu.edu.
[1]Johns Hopkins University (JHU), [2]University of Pennsylvania, [3]Loughborough University.

able to leverage the out-of-distribution data; that is, it fails to quantify the magnitude of actual transfer. Therefore, measures of predictive performance, such as accuracy, are inadequate when evaluating out-of-distribution learning, if we are to understand which properties of these algorithms are doing which work. The evaluation criteria could instead compare performance within an algorithm when obtaining and not obtaining additional out-of-distribution data. While many authors have proposed criteria such as forward and reverse transfer, the theoretical motivation for choosing such criteria were lacking [13–16]. To give one concrete example from our own work, the efficient lifelong learning algorithm (ELLA) [13] specifies a clear objective function, provides convergence guarantees, and works well for shallow models with sufficiently compact task distributions (i.e., tasks that are sufficiently similar). However, ELLA does not provide any guarantees on what it will converge to, nor how it will generalize to more diverse tasks [17] or deeper models [18].

Second, the literature remains confused on a number of central issues. For example, what counts as OOD learning: must the learner perform well for any possible new task in an environment (as in meta-learning and domain generalization [19]), or a specified new task (as in domain adaptation [20] and covariate shift [21])? What differentiates online, continual, and lifelong learning, and how are they related to OOD learning? Do lifelong learners have task information in training and/or testing [10], or must they also infer the task itself [22]? And does it count as lifelong learning if computational complexity is constant [23], scales quasilinearly [24], or quadratically [25] with sample size?

We therefore revisit the foundations of learning [26], and endeavor to update them in three ways. First, we define a generalized learning task in such a way that includes both in-distribution and out-of-distribution (§ 2). Specifically, **we make a simple change to the classic in-distribution definition of a learning task: we no longer implicitly assume that the evaluation distribution that our learner will face at test/deployment time is the exact same distribution from which the training data are assumed to be sampled** (Figure 1, left).

Second, we introduce **learning efficiency**, which quantifies how much a given learner is able to learn a task by leveraging data (Section 3.2), regardless of whether it is in- or out-of-distribution data. Learning efficiency is closely related to, and inspired by, relative efficiency from statistical estimation [27]. Whereas relative efficiency quantifies the relative number of samples required by one learner to achieve the same error as another, learning efficiency quantifies the relative number of samples required by one learner relative to the same learner that obtains other (potentially out-of-distribution) data.

Third, we formally define several notions of **out-of-distribution learnability**, including weak, strong, non-uniform, and consistency, to complement and extend in-distribution variants of probably almost correct (PAC) learnability [28], non-uniform learnability [29], and consistency [27] (Section 3). Our definitions continue Valiant's work and formalizes Tom Mitchell's colloquial definition of learning *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."* [30]. We then prove the relationship between each of these generalized notions of learnability. There are three conceptual differences between our definition of learning and previous definitions. First, in previous definitions, the evaluation distribution and training data distribution were (often implicitly) assumed to be the same. Here, those two distributions need not be related (though they must be related to solve the problem). Second, our explicit goal is to *improve* performance with (data), rather than, say, achieve (Bayes) optimal performance—a much more modest aspiration. Third, we do not require large sample sizes, rather, our definition includes zero-shot and few-shot learning. Note that these three relaxations of existing formal definitions of learning improve the alignment between machine and natural (biological) learning, which partially motivated this work.

Fourth, we leverage our definition of learning, learnability, and learning efficiency to formalize, quantify, and hierarchically organize learning in multiple distinct OOD learning scenarios, including transfer, multitask, meta, continual, and lifelong learning (Section 5 and Figure 1, right). Doing so required formalizing some conventions that are not yet universally agreed upon. While some readers may disagree with some of the choices we made in defining this hierarchy, our main point is not the particular hierarchy *per se*. Rather, our primary intention is to illustrate the flexibility of our formalism; specifically, that it provides a single coherent lens through which essentially any learning scenario (including both in-distribution and out-of-distribution learning) may be evaluated, much like generalization error provides for in-distribution learning (Figure 2).

Finally, Thrun pointed out in 1996 that lifelong learning is essential for human learning [10]. Moreover, it seems essential for biological learning more generally, not just human learning [31]. For example, honey bees with only ≈1 million neurons can learn 'same versus different' tasks, and moreover, they can perform zero-shot cross-model transfer learning [32]. Other species with larger brains can perform all sorts of tasks that remain outside the realm of modern ML/AI [33]. We believe that OOD learning is the key capability that biological learning agents leverage to achieve natural intelligences that surpasses modern ML/AI [34]. And therefore, we hope our contribution will help bridge the gap between biological and machine learning and intelligence, which was one of Valiant's originally stated goals, as implied by the first sentence of his seminal paper: "Humans appear to be able to learn new concepts without needing to be programmed explicitly in any conventional sense." [28]

2 A proposed unified framework for learning

Here we formally introduce our learning framework, which generalizes the canonical in-distribution learning framework proposed by Glivenko [35] and Cantelli [36] nearly 100 years ago (and was further refined by Vapnik and Chervonenkis [37] and [28] 50 years

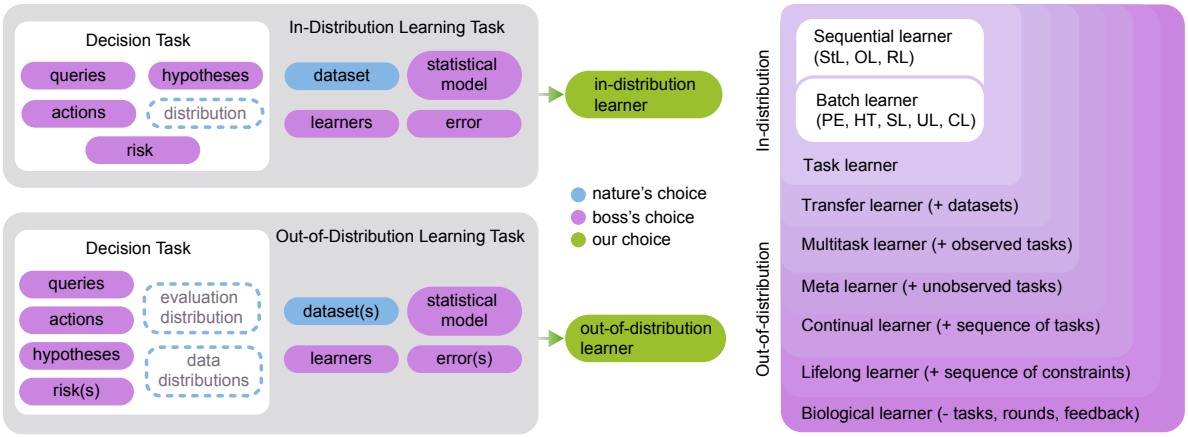

Figure 1: (Left) Decision tasks (top) are composed of five components, and the goal is to choose a hypothesis based on the known distribution that minimizes risk. In an in-distribution learning task (middle), the distribution is not available, so a feasible in-distribution learner must leverage a dataset, to find a hypothesis that minimizes error under an assumed statistical model. In out-of-distribution learning tasks (bottom), the distribution over queries need not be about the assumed distribution of the data, and there may be multiple datasets, risks, and errors. Each component is provided by one of three different actors: nature, the boss, or the machine learning practitioner. (Right) Schematic illustrating the nested nature of learning problems. PE = point estimation; HT = hypothesis testing; SL = supervised learning; UL = unsupervised learning; CL = causal learning; StL = streaming learning; OL = online learning; RL = reinforcement learning.

IN / OUT	BATCH / STREAMING	LEARNING SCENARIO	UNLABELED DATA	LABELED DATA	>1 DATASETS	>1 TASKS	UNOBSERVED TASKS	STREAMING TASKS	STREAMING PRIORITIES & RESOURCES	STREAMING DATA	COMPUTATIONAL CONSTRAINTS
in distribution	batch	point estimation	Y								
		unsupervised learning	Y								
		hypothesis testing	Y	O							
		supervised learning	Y	Y							O
		causal learning	Y	Y							
	streaming	streaming learning	Y	O						Y	Y
		online learning	Y	O						Y	
		reinforcement learning	Y	Y						Y	O
out of distribution	batch	transfer learning	Y	O	Y					O	O
		multitask learning	Y	O	Y	Y				O	O
		meta-learning	Y	O	Y	Y	Y			O	O
	streaming	continual learning	Y	O	Y	Y	Y	Y		O	Y
		lifelong learning	Y	O	Y	Y	Y	Y	Y	O	Y
		biological learning	Y	O	Y	n/a	n/a	n/a	Y	Y	Y

Figure 2: The characteristics of different learning paradigms. Y: present, O: optional, n/a: not applicable.

ago) to include out-of-distribution learning problems, such as transfer, multitask, meta, continual, and lifelong learning. We follow the notation established by Shalev-Shwartz et al. [26] and Fokoué [38] for in-distribution learning, departing as appropriate to generalize to include out-of-distribution learning. Our key departure (and the novel contribution in this section) is the relaxation of the assumptions on the distribution governing the evaluation and training data in the learning problem.

2.1 The decision problem Learning problems, as will be defined below, leverage data to solve decision problems. We therefore first formalize decision problems ([27], Chapter 1.3). Note that decision problems, as described here, do not include any data, rather, they are simply mathematical formalizations of a goal.

- **Query space** $\mathcal{X}$ is the set of questions, inputs, or test examples given to the agent, $x \in \mathcal{X}$.
- **Action space** $\mathcal{Y}$ is the set of potential actions, or predictions, to be returned given a query $x \in \mathcal{X}$.
- **Hypothesis Space** $\mathcal{H} \subseteq \{h \mid h : \mathcal{X} \mapsto \mathcal{Y}\}$ is a set of hypotheses that an agent considers, which is a (potentially strict) subset of all possible functions mapping $\mathcal{X}$ to $\mathcal{Y}$.
- **Evaluation Distribution** $P = P_{X,Y}$. We assume that the query-action pairs, (x, y) are realizations of query- and action-valued random variables, (X, Y), which are drawn from some evaluation distribution $P = P_{X,Y}$ which is an element of a set of possible evaluation distributions $P_{X,Y} \in \mathcal{P}_{X,Y}$.
- **Risk** $R : \mathcal{H} \times \mathcal{P}_{X,Y} \mapsto \mathbb{R}_{\geq 0}$. A risk function takes a hypothesis and a distribution on the random variable (X, Y) and evaluates the performance of that hypothesis against the distribution. Often, implicit in the risk functional, is a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}_{\geq 0}$, and the risk is defined as the expected loss with respect to $P_{X,Y}$: $R_{X,Y}(h) = \mathbb{E}_{X,Y}[\ell(h(X), Y)] = \int_{\mathcal{X},\mathcal{Y}} \ell(h(X), Y) dP_{X,Y}$.

Given these components, we can define a **decision problem**:

Definition 2.1 (Decision Problem). *A decision problem has the form*

$$\begin{array}{ll} \text{minimize} & R_{X,Y}(h) \\ \text{subject to} & h \in \mathcal{H} \end{array} .$$

Let $h^\diamond$ denote the **decision problem optimal hypothesis**, that is, the hypothesis that minimizes the objective function subject to the constraints imposed by $\mathcal{H}$ in this decision problem. Let $R^\diamond = R(h^\diamond)$ denote the **decision problem optimal risk**, that is, the minimum subject to the constraints imposed by $\mathcal{H}$ in this decision problem. Note that $h^\diamond$ need not be unique, and that $h^\diamond$ nor $R^\diamond$ need not exist (in which case we could be looking for an infimum rather than a minimum). Let R^* denote the **Bayes optimal risk** for a given decision problem without constraints on $\mathcal{H}$. The **excess risk** or **model error** is defined by $R^\diamond - R^*$ [39].

In decision tasks can be thought of as involving three different players: nature, the boss, and us. Nature provides the true distribution, and "the boss" provides the other elements mentioned above. Our goal is to choose a good hypothesis (Figure 1, left top).

2.2 The learning problem Solving decision problems depends on knowing the true distribution, $P_{X,Y}$. In learning problems, the true distribution is partially or completely unknown, and so, a learner must leverage data to estimate hypotheses. A **learner** uses n data points to provide a good guess for h in the sense of Equation (2.1). This is the learning task. Defining it formally requires a few additional concepts.

- **Data Space** $\mathcal{S}^*$. Let $s \in \mathcal{S}$ denote a data sample. Data samples can be query-action pairs in supervised learning problems, that is, $\mathcal{S} = \mathcal{X} \times \mathcal{Y}$. More generally, the data need not be query-action pairs; for example, consider unsupervised learning in which case samples might purely be in $\mathcal{X}$. Our data space is the union of all possible sequences of those pairs, that is, $\mathcal{S}^* = \bigcup_{n=0}^{\infty} \mathcal{S}^n$. Thus, $\mathcal{S}^*$ is the set of all n-length lists of $\mathcal{S}$-valued data, for any $n \in \mathbb{N}$.
- **Statistical Model** $\mathcal{P} = \mathcal{P}_{X,Y,\mathbf{S}} = \mathcal{P}_{X,Y} \otimes \mathcal{P}_{\mathbf{S}}$, is a collection of admissible distributions. Recall from the definition of a decision problem, we assumed that query-action pairs were sampled according to some evaluation distribution $P_{X,Y} \in \mathcal{P}_{X,Y}$. In decision problems, the evaluation distribution is known, and therefore, hypotheses could be evaluated with respect to the true evaluation distribution. In learning problems, the evaluation distribution is unknown, and therefore, a learner (defined below) leverages data to obtain a good approximate hypothesis. We assume that a training dataset, $\boldsymbol{s}_n = \{s_1, \ldots, s_n\} \in \mathcal{S}^n$, is sampled according to some training dataset distribution $P_{\mathbf{S}_n}$, which is an element of $\mathcal{P}_{\mathbf{S}_n}$, and $\mathcal{P}_{\mathbf{S}} = \cup_n \mathcal{P}_{\mathbf{S}_n}$, and $P_{\mathbf{S}} \in \mathcal{P}_{\mathbf{S}}$. For in-distribution problems, there is a strong (often implicit) assumption on the relationship between each $P_{\mathbf{S}}$ and $P_{X,Y}$: specifically, it is assumed that the queries are *about* $P_{\mathbf{S}}$, that is, queries are about the distribution that we assume the training data are sampled from. In contrast, **in out-of-distribution learning, there need not be any assumption whatsoever on the relationship between these two distributions:** $P_{\mathbf{S}}$ **and** $P_{X,Y}$.
- **Learners** $\mathcal{F} \subseteq \{f : \mathcal{S}^* \times \Lambda\} \mapsto \mathcal{H}'$, where f is a learner, and $\mathcal{H}'$ is the set of hypotheses that f can yield. Note that in general, $\mathcal{H}'$ and $\mathcal{H}$—the set of hypotheses that f can find and the set of hypotheses that are allowed by the decision problem, respectively—need not be the same.[1] The first input to the learner is the data which the learner will utilize to learn a hypothesis, $\boldsymbol{s}_n \in \mathcal{S}^*$. The second input is the hyperparameter, $\lambda \in \Lambda$, which can incorporate prior knowledge, initial conditions, and potentially other kinds of side information [40]. We will typically suppress the

[1] For example, the decision problem may be looking for any linear function, but the learner we use may only search over linear functions of the form $h(\boldsymbol{x}) = \boldsymbol{x}^\mathsf{T} \boldsymbol{\Sigma}^{-1} \boldsymbol{\delta}$, i.e., Fisher's linear discriminant.

Λ-valued inputs for notational brevity, so we write $f(\mathbf{S}_n) = \hat{h}_n$.

- **Error** $\mathcal{E} : \mathcal{F} \times \mathcal{S}^* \times \mathcal{P}_{X,Y,\mathbf{S}} \times \mathcal{R} \mapsto \mathbb{R}$, where $\mathcal{R}$ is the set of possible risk functions. Because the error of the learner depends on the data, which we assume are realizations of random variables, the risk is also a random variable. We define **error**, $\mathcal{E}_f(\mathbf{S}_n)$ of a learner f on dataset $\mathbf{S}_n$ as

$$\mathbb{E}_{\mathbf{S}_n \sim P_{\mathbf{S}_n}}[R(\hat{h}_n)] = \int_{\mathcal{S}^n} R(f(\mathbf{S}_n)) dP \ , \tag{2.1}$$

or the above minus the Bayes or class optimal risk. When risk is defined as expected loss, the above equation can be further expanded as follows

$$\mathbb{E}_{\mathbf{S}_n \sim P_{\mathbf{S}_n}}[\mathbb{E}_{P_{X,Y}}[\ell(f(\mathbf{S}_n)(X), Y)]] = \int_{\mathcal{S}^n} \int_{\mathcal{X},\mathcal{Y}} \ell(f(\mathbf{S}_n)(X), Y) dP_{X,Y} dP_{\mathbf{S}_n} \ , \tag{2.2}$$

where the internal integral is with respect to the evaluation distribution $P_{X,Y}$, and the external integral is with respect to the dataset distribution $P_{\mathbf{S}_n}$. Combining all of the introduced components, we can now define a **learning task**.

Definition 2.2 (Learning Task). *Given a query space $\mathcal{X}$, action space $\mathcal{Y}$, hypothesis class $\mathcal{H}$, risk function R, and n samples $\mathbf{S}_n$ drawn according to some true but unknown data distribution $P_{\mathbf{S}_n}$, and query-action pairs drawn from some true but unknown evaluation distribution $P_{X,Y}$, the task t is to choose a learner $f \in \mathcal{F}$ that learns a hypothesis $\hat{h}_n$ that minimizes error $\mathcal{E}$:*

$$\begin{array}{ll} \text{minimize} & \mathcal{E}_f^t(\mathbf{S}_n) \\ \text{subject to} & f \in \mathcal{F} \end{array} .$$

The superscript t on the error $\mathcal{E}$ indicates that this is the error *with respect to task* t; it does not indicate that the learner gets as input all the task information, which would include the true but unknown distribution $P_{X,Y,\mathbf{S}}$. However, the error function $\mathcal{E}$ must know the true distribution in order to compute the expected risk. In learning problems, nature still chooses the distribution—both the evaluation and data distributions—but those distributions are not provided to the learner. Instead, nature provides a dataset (or multiple datasets). The boss provides the remaining components. Our goal is to choose a good learner (Figure 1, left bottom). Below, we will consider scenarios with multiple tasks, there we will alternate notations to sometimes let t denote the task, or t index a specific task, and we hope it is clear by context.

3 Generalized definitions of learnability

A learning task defines the goal of a learner. Given such a definition, it is natural to wonder whether a particular learner achieves that goal, i.e., whether a task is *learnable* for a learner. More generally, we may desire to know the extent to which a learner achieves that goal. In classical in-distribution learning theory, there are many formal, complementary definitions of learning, including weak learnability [41], strong learnability [42], non-uniform learnability [26], and (universal) consistency [27]. None of these definitions of learning, however, are sufficiently general to account for all the modern learning paradigms we study in ML/AI, including transfer, multitask, meta, continual, and lifelong learning. For that reason, here we extend those definitions of learning to include the various out-of-distribution learning scenarios described above.

3.1 The generalized learning problem

In generalized learning problems, there may be multiple datasets, tasks, learners, etc. For this reason, we first augment the above components of a learning task to be able to account for this.

- **Learning Setting** $b = (\mathcal{X}, \mathcal{Y}, \mathcal{S}^*, \mathcal{P}, \mathcal{H}, R, \mathcal{F}, \mathcal{E})$, where $b \in \mathcal{B}$, and each of the objects in the set b is defined above. In-distribution learning tasks were characterized by a single setting, and therefore, explicitly defining a setting was unnecessary. However, in out-of-distribution learning, we often have tasks with multiple settings. Note that $\mathcal{H}$ implicitly defines $\mathcal{X}$ and $\mathcal{Y}$, both $\mathcal{P}$ and $\mathcal{F}$ are often set to be maximally general, and $\mathcal{E}$ is typically the expectation over the training data, and so a setting can often be described simply by the data space, risk, and feasible hypotheses: $(\mathcal{S}^*, R, \mathcal{H})$.
- **Data Space** $\mathcal{S}^* \leftarrow (\mathcal{S}^*, \mathcal{D})$. In OOD learning, there can be multiple datasets, so datasets become pairs $(\mathbf{S}^k, d^k)$ of the actual data and potentially a "dataset descriptor" which includes an index to uniquely identify each dataset (more generally, they could also include additional side information). We will assume in out-of-distribution learning problems that we have a set of J datasets, $\mathbf{S}_n = \{\mathbf{S}^1, \mathbf{S}^2, \ldots, \mathbf{S}^J\}$, where each $\mathbf{S}^j$ has sample size n_j, and $n = \sum_{j=1}^J n_j$, and $\boldsymbol{n} = (n_1, n_2, \ldots, n_J)$. A key departure from in-distribution learning is that each dataset may be associated with any number of tasks.
- **Query Space** $\mathcal{X} \leftarrow (\mathcal{X}, \mathcal{B})$. Because there can be multiple settings, queries can specify which setting it is associated with (though it also may not). For example, a query could be "what is x in

setting b?" when there are multiple different settings, or it could simply be "what is x?" without specifying a setting (e.g., in task-agnostic lifelong learning [43, 44]).

- **Statistical model** $\mathcal{P} = \mathcal{P}_{X,Y,\mathbf{S}}$, is still a collection of distributions, but the distribution over data $P_{\mathbf{S}}$ can be a mixture over datasets. More specifically, each dataset $\mathbf{S}^j$ is distributed according to some distribution P^j, $j = 1, \ldots, J$, and $P_{\mathbf{S}} = \sum_j \pi_j P^j$, where π_j is the mixture coefficient for dataset j.
- **Learner** $\mathcal{F} \subseteq \{f : \mathcal{S}^* \times \Lambda \times \mathcal{B}\} \mapsto \mathcal{H}'$. OOD task learners can also operate on a setting for a given task.
- **Learning scenario** is defined by a set of tasks $\mathcal{T}$ (the **environment**) along with a set of J datasets of size $\boldsymbol{n}$, where J and $|\mathcal{T}|$ can both be ≥ 1.

Given all the above machinery, we can now define a generalized notion of a learning task which includes out-of-distribution learning scenarios, including transfer, multitask, meta, continual, and lifelong learning.

Definition 3.1 (Generalized learning task). *Given an environment of tasks $\mathcal{T}$, and sample sizes $\boldsymbol{n}$, drawn according to some true but unknown data distribution $P_{\mathbf{S}}$, and query-action pairs drawn from some true but unknown evaluation distributions $P^t_{X,Y}$ for each task, and a weight for each task w_t corresponding to the extent the learner prioritizes task t, the generalized task is to choose a learner $f \in \mathcal{F}$ that learns a hypothesis $\hat{h}_n$ that minimizes error $\mathcal{E}$. Letting $\mathcal{E}^{\mathcal{T}}_f(\mathbf{S}_n) = \sum_{t \in \mathcal{T}} w_t \mathcal{E}^t_f(\mathbf{S}_n)$,*

$$\begin{array}{ll} \text{minimize} & \mathcal{E}^{\mathcal{T}}_f(\mathbf{S}_n) \\ \text{subject to} & f \in \mathcal{F} \end{array} .$$

It is hopefully apparent that the generalized learning task (Definition 3.1) introduced here, is indeed a generalization of the classic learning task (Definition 2.2 from Section 2). Note that nowhere have we specified that any data samples were independent or identically distributed, so the above framework can also incorporate more general scenarios. For example, the above formalism can incorporate "task agnostic" scenarios in which there are multiple tasks, but the learner and/or hypothesis does not know, for a given sample, which task it is associated with [43, 44]. This contrasts with the simpler "task aware" scenarios in which the learner and hypothesis both know, for each sample, which task it is associated with, and all the relevant setting information associated with each task. These two scenarios correspond to opposite ends of a spectrum of "task semi-aware" scenarios, in which the learner or hypothesis sometimes does not know the task associated with a given sample, or only has partial information about some of the tasks. Extending the framework further, for example, to incorporate online learning scenarios, does require a modification that the learners operate also on previous hypotheses. Doing so could also enable "open world learning" [45], which is a kind of online task semi-aware scenario.

3.2 Learning Efficiency

With the definition of a generalized learning task in hand, a next natural question is whether a given learner f has learned. Here, we build what we believe to be the simplest (and weakest) notion of learnability, which we term *learning efficiency*. Learning efficiency builds on the notion of relative efficiency from statistical estimation [27]. In statistics, the relative efficiency of two different consistent estimators f and f' is the ratio of their variances (which is equal to the ratio of their generalization errors under certain assumptions). In contrast, learning efficiency evaluates the error of one particular learner f, based on only having some target data $\mathbf{S}^A$, versus also having some source data $\mathbf{S}^B$.

Definition 3.2 (Learning Efficiency). *The learning efficiency* (LE) *for task* t *with learner* f *is a function* $\mathsf{LE}_f^t : \mathcal{S}^* \times \mathcal{S}^* \to \mathbb{R}$ *is defined by*

$$\mathsf{LE}_f^t(\mathbf{S}^A, \mathbf{S}^B) = \frac{\mathcal{E}_f^t(\mathbf{S}^A)}{\mathcal{E}_f^t(\mathbf{S}^{A\cup B})},$$

where each expectation in the error is taken with respect to the corresponding dataset distribution, i.e. the numerator expectation is taken with respect to the distribution governing $\mathbf{S}^A$*, and the denominator expectation is taken with respect to the distribution governing* $\mathbf{S}^{A\cup B}$*.*

Learning efficiency is therefore a function of S^A and S^B for any given samples sizes, n and m, respectively (including if n and/or $m = 0$). For a given distribution, one could therefore plot the learning efficiency two-dimensional surface, which would completely characterize the learning efficiency for the given task. An increase in learning efficiency corresponds to f learning more from $\mathbf{S}^B$ for task t.

Implicit in this definition is that the f in the numerator and the f in the denominator are operating on the same hyperparameter, λ. In some cases, the hyperparameter λ could be a function of the sample size, for example, if sample size is small, λ penalizes more. In that case, we would write λ_n as the penalty for sample size n, and we would require that the sequence $\boldsymbol{\lambda} = \lambda_1, \lambda_2, \ldots$ would be the same sequence in the numerator and denominator.

Evaluating this ratio requires training and evaluating the exact same f *twice*: once with only $\mathbf{S}^A$, and once also including $\mathbf{S}^B$. If this ratio is bigger than 1 (or, equivalently, its log is positive), then we say that f *learned* from $\mathbf{S}^B$—over and above that which f learned via simply $\mathbf{S}^A$—with respect to task t. Learning efficiency thus allows us to measure the extent to which f leverages the data $\mathbf{S}^B$ to improve performance on task t. This leads us to our first, and the simplest, definition of whether an agent or function f has learned: f **learns about** $P_{X,Y}$ **from** S^B **whenever** $\log \mathsf{LE}_f^t(\mathbf{S}^A, \mathbf{S}^B) > 0$. Note that this immediately implies a definition of *forgetting*: $\log \mathsf{LE}_f^t(\mathbf{S}^A, \mathbf{S}^B) < 0$. While interference and forgetting is often discussed in articles about both natural and artificial intelligence[25, 46], we have as yet seen any formal definition of forgetting in the literature .

We note that these definition of learning and forgetting are "global", in the sense that, in expectation, the risk of the hypothesis after learning improved over the risk prior to the additional data. In other words, these definitions perform an average over the entire space, rather than being restricted to any one part of the space. Note that because this is an average, one can also locally learn even if one does not globally learn. Imagine for example, that the induced hypothesis improved its risk in some subspace of the query space $\mathcal{X}' \subsetneq \mathcal{X}$, but got worse (aka, forgot) in another subspace $\mathcal{X}'' \subsetneq \mathcal{X}$, where $\mathcal{X}' \cap \mathcal{X}'' = \emptyset$. In such a scenario, depending on $P_{X,Y}$, it could be that expected risk (i.e. the risk averaged over the entire space) got worse, in which case, overall learning efficiency would be log negative. So, in this scenario, f locally learned and globally forgot.

As noted above, in learning efficiency, there is no minimum sample size. Rather, one can compute learning efficiency for any $n, m \geq 0$. Thus, zero-shot and few-shot learning are perfectly fine notions of learning that can be theoretically and empirically investigated via studying learning efficiency. This is in stark contrast to in-distribution definitions of learning, all of which require a sufficiently large sample size to ascertain whether f has learned.

3.3 Out-of-Distribution Learnability

There are several well established definitions of learnability available in the in-distribution literature, including including weak [41], strong [42], and non-uniform learnability [26], as well as (universal) consistency [27]. In this subsection, we illustrate how to generalize these notions of learnability to include out-of-distribution learning. Note that some definitions of in

distribution learnability include a computational complexity constraint. Following Shalev-Shwartz et al. [26], we have decided to ignore such constraints when defining out-of-distribution learnability.

Weak learnability, in classical in-distribution learning, informally corresponds to performing marginally better than chance with a sufficiently large sample size. In the in-distribution learning setting, it is implicitly assumed that the evaluation and training distributions are the same. In weak out-of-distribution learnability, we simply relax that assumption. For notational simplicity, given a pair of datasets, $\mathbf{S}^A$ and $\mathbf{S}^B$, with sample sizes n and m respectively, we denote the hypothesis learned leveraging only $\mathbf{S}^A$ by $\hat{h}_n$, and the hypothesis learned by leveraging both $\mathbf{S}^A$ and $\mathbf{S}^B$ by $\hat{h}_{n,m}$. We also let $\mathbf{S}_{n,m} = \mathbf{S}^A \cup \mathbf{S}^B$.

Definition 3.3 (Weak OOD Learnability). *A model $\mathcal{P}_{X,Y,\mathbf{S}}$ is weakly OOD learnable with target sample size n if there exists a learner f such that for all $P_{X,Y,\mathbf{S}} \in \mathcal{P}_{X,Y,\mathbf{S}}$ and $\delta > 0$, there exists $M(\delta) \in \mathbb{N}$ such that for $m \geq M$ we have*

$$\mathbb{P}_{\mathbf{S}_{n,m}}[R_{XY}(\hat{h}_{n,m}) < R_{XY}(\hat{h}_n)] \geq 1 - \delta.$$

The above definition states that a model $\mathcal{P}_{X,Y,\mathbf{S}}$ is (n, m, δ)-weakly OOD learnable if, given a source sample size $m > M$, and a target sample size n, f is able to improve performance (e.g., reduce generalization error over the task data only hypothesis $\hat{h}_n$) with probability of at least $1 - \delta$. Or, stated a bit more succinctly, f learns from $\mathbf{S}_{n,m}$ about $P_{X,Y}$ when f's performance $R_{X,Y}$ improves due to data $\mathbf{S}_{n,m}$ with at least probability $1 - \delta$. The statistical model $\mathcal{P}_{X,Y,\mathbf{S}}$ can be as simple as a singleton model (e.g., the standard normal Gaussian) as complicated as the universal model (e.g., any conceivable distribution defined on the same domain). When $\mathcal{P}_{X,Y,\mathbf{S}}$ is the universal model, we say that f universally weakly learns from $\mathbf{S}_n$ about $P_{X,Y}$. The key to a model being weakly OOD learnable is that the source data $\mathbf{S}^B$ has some information about the evaluation distribution $P_{X,Y}$ that complements the information about the evaluation distribution in the target data $\mathbf{S}^A$.

By construction, weak OOD learnability is closely related to what is typically called 'weak learnability', which we refer to hereafter as 'weak in-distribution learnability' to avoid confusion. There are several key differences between weak OOD learnability and weak in-distribution learnability. First, implicit in weak in-distribution learnability, the queries were about the assumed data distribution. In OOD learning, we have relaxed that assumption: the data distributions and evaluation distributions need not even be related, and the data can be from a mixture of some informative and some uninformative distributions. Second, we have made no independent and identically distributed assumptions in stating this definition, $P_{\mathbf{S}}$ can be quite flexible. Third, rather than doing better than chance as required for weak in-distribution learnability, in weak OOD learnability, the source data S^B must enable f to perform better than it would have with only target data S^A. This change is motivated by the fact that every learner f has some *inductive bias which comes from somewhere*. That somewhere must be abstracted from prior knowledge, experiences, and data. In that sense, the question of whether f learned from data must address f's prior knowledge about $P_{X,Y}$, and so we encode it with the hypothesis that f learns using some data, $\mathbf{S}_n$. Doing so motivates that weak OOD learnability is a function of two sample sizes: one for the target data and one for the source data. Finally, weak OOD learnability is with respect to a statistical model $\mathcal{P}_{X,Y,\mathbf{S}}$, rather than a hypothesis class $\mathcal{H}$. This change is for convenience.

previous assumptions

Strong learnability, in classical in-distribution learning, informally corresponds to performing arbitrarily well with a sufficiently large sample size. Just like weak in-distribution learning, for strong in-distribution learning, it is assumed that the evaluation and training data distributions are the same. We relax that assumption here as well to obtain strong OOD learnability.

Definition 3.4 (Strong OOD Learnability). *A model $\mathcal{P}_{X,Y,\mathbf{S}}$ is strongly OOD learnable with target*

sample size n if there exists a learner f such that for all $P_{X,Y,\mathbf{S}} \in \mathcal{P}_{X,Y,\mathbf{S}}$ and $\epsilon, \delta > 0$, there exists $M(\epsilon, \delta) \in \mathbb{N}$ such that for $m \geq M$, we have

$$\mathbb{P}_{\mathbf{S}_{n,m}}[R_{XY}(\hat{h}_{n,m}) - R^* < \varepsilon] \geq 1 - \delta.$$

For in-distribution consistency, we need to be able to perform arbitrarily well with arbitrarily high probability given enough data [27], which is exactly what strong learnability is if one ignores the computational complexity issues that are commonly included (but ignored here). Thus, OOD consistency is equivalent to strong OOD learnability as defined here. Universal OOD consistency corresponds to the setting in which the source data goes to infinity, but the target data is a fixed, finite amount, and the expected risk approaches Bayes error for any conceivable distribution. **Non-uniform OOD learnability** is exactly the same as strong OOD learnability, except that J also depends on the distribution $P_{X,Y,\mathbf{S}}$ in that case.

minimum possible risk

3.4 Mitchell's definition of learning Recall Tom Mitchell's definition of learning: *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."*. We view this quote as a kind of out-of-distribution weak learner claim [42]. Let's define each of Mitchell's terms using our notation:

- *computer program*: is the learner, f,
- *experience*: is the dataset, $\mathbf{S}_n$,
- *task*: is the task, t, Definition 2.2,
- *performance measure*: is risk, $R_{XY}(\hat{h}_n)$.

Mitchell's claim is therefore that f learns from $\mathbf{S}_n$ on task t when $R_{XY}(\hat{h}_n)$ improves, which is exactly equivalent to our definition of weak OOD learnability in Definition 3.3.

4 Theoretical Results In this section we establish the relationship between our newly introduced generalized notions of learnability: positive transfer, weak, strong and non-uniform learnability, and consistency.

Proposition 4.1 (Strong implies weak). *For an arbitrary task t, if we have strongly OOD learned at target sample size n, then we have weakly OOD learned at target sample size n.*

Proof. The proof follows very quickly from definitions. To have strongly OOD learned means that we can get arbitrarily to optimal performance with arbitrarily high probability. To weakly OOD learn is to perform better than base performance $R_{XY}(\hat{h}_n)$ with arbitrarily high probability. Performing arbitrary well certainly implies performing better than base performance. Thus strong OOD learning implies weak. ■

Proposition 4.2 (Strong implies non-uniform). *For an arbitrary task t, if we have strongly OOD learned at sample size n, then we non-uniformly OOD learned at sample size n.*

Proof. The proof also follows quickly from definitions. To have strongly OOD learned means that we can get arbitrarily close to optimal performance with arbitrarily high probability for all the distributions in the task model given a certain number of data points. To non-uniformly OOD learn means that we can get arbitrarily close to optimal performance with arbitrarily high probability for all the distributions in the task model given a certain number of data points that depends on the distribution. It is clear then that strong implies non-uniform since the number of data points required for the performance bound in strong OOD learning implies that this number of data points gives us the desired performance bound for all the distributions. This implies non-uniform OOD learning. ■

Theorem 4.3 (Positive transfer is weaker than weak OOD learning). *For an arbitrary task t (with source data $\mathbf{S}^B$), if f weakly OOD learns from $\mathbf{S}^B$, then f also positively transfers from $\mathbf{S}^B$. However, the converse is not always true. Formally, we have*

$$\mathbb{P}_{\mathbf{S}_{n,m}}[R_{XY}(\hat{h}_{n,m}) < R_{XY}(\hat{h}_n)] \geq 1 - \delta. \implies \log \mathsf{LE}_f^t(\mathbf{S}_n^A, \mathbf{S}_m^B) > 0.$$

Thus, the class of positive transfer learning problems is bigger than the class of weakly OOD learning problems.

In words, there are some tasks where there exists an $f \in \mathcal{F}$ such that a specific f exhibits positive transfer, even though it cannot weakly OOD learn, for some sample sizes n, m. In contrast, if f has weakly OOD learned with those sample sizes, then it has also positively transferred.

Proof. Assume that we have weakly OOD learned for the given task t with source data $\mathbf{S}^B$ with sample size m. This implies that $R_{n,m} = R_{X,Y}(\hat{h}_{n,m}) < R_{X,Y}(\hat{h}_n) = R_n$ with probability 1 (since δ can be made arbitrarily small). Thus, the expected value of $R_{n,m}$ is less than the expected value of R_n. Recall that transfer efficiency is a ratio of errors, or a ratio of expected risks. Thus, we get

$$\frac{\mathcal{E}_f(\mathbf{S}_n)}{\mathcal{E}_f(\mathbf{S}_{n,m})} = \frac{\mathbb{E}_f(R_n)}{\mathbb{E}_f(R_{n,m})} > 1.$$

This implies then that the log of that ratio is greater than 0, which is the desired result. Thus weak OOD learning implies transfer learning.

However, having positively transfer does not imply having weakly OOD learned. In essence, this is because positive transfer is about having the average risk for the transfer learner (utilizing both task data and out-of-task data) is less than the average risk for the task learner (utilizing only the task data). However, as this is an average, this does not preclude the transfer learner risk being larger (worse) with a certain probability (i.e., forgetting). Weak OOD learning requires that the transfer learner's risk be less than the task learner risk with arbitrarily high probability. Here is an example scenario demonstrating this fact.

Consider the same $0-1$ classification task from the proof of theorem 4.4, with the same algorithm f. Let the evaluation distribution $P_{X,Y}$ be $P_{0.8}$ and let the source data distribution P_s be a coin flip where the data is drawn iid according to $P_{0.7}$ with probability 0.5 or otherwise drawn iid according to $P_{0.1}$. Assume now that for this task we have $n = 0$ target task data, i.e. $\mathbf{S}^A = \varnothing$, and $\mathbf{S}^B$ is drawn according to P_s. f then outputs hypotheses of the form $\hat{h}_z(x) = \mathbb{I}(x \leq z)\mathbb{I}(y = 0) + \mathbb{I}(x > z)\mathbb{I}(y = 1)$. Depending on the coin flip, the best possible hypothesis that f can output will either be $\hat{h}_{0.7}$ or $\hat{h}_{0.1}$. Under very high m, with probability one, the outputted hypothesis $\hat{h}_{n,m}$ will be very close to either $\hat{h}_{0.1}$ or $\hat{h}_{0.7}$. Thus, with probability 1 under very high m, the expected risk is approximately

$$\begin{aligned}
\mathbb{E}_{P_{\mathbf{S}_{n,m}}}[R_{X,Y}(\hat{h}_{n,m})] &\approx 0.5 \int_0^1 \mathbb{I}(\hat{h}_{0.7}(x) \neq h^*_{0.8}(x))dx + 0.5 \int_0^1 \mathbb{I}(\hat{h}_{0.1}(x) \neq h^*_{0.8}(x))dx \\
&= 0.5 \cdot 0.1 + 0.5 \cdot 0.7 \\
&= 0.4,
\end{aligned}$$

where h^*_z is the Bayes hypothesis for P_z. This is better than the task learner $\hat{h}_n$ which is outputted by f with $n = 0$ and simply flips a coin for any given x and outputs 0 or 1 according to the result. The expected risk of this h_n is 0.5. Thus, we have positive transfer in this case. However, no matter how high we make m, we can never weakly OOD learn with target sample size 0. This is because no matter how

high we make m, the outputted hypothesis, with probability 0.5, will be a hypothesis formed according to $P_{0.1}$. In terms of risk, the best possible hypothesis that we can get from this data is $\hat{h}_{0.1}$, which has a risk of 0.7. This 0.7 expected risk is a lower bound. This is higher than 0.5 and thus we cannot, with arbitrarily high probability, have $R_{XY}(\hat{h}_{n,m}) < R_{XY}(\hat{h}_n)$. In fact, we have in our scenario that

$$\mathbb{P}_{\mathbf{S}_{n,m}}[R_{XY}(\hat{h}_{n,m}) < R_{XY}(\hat{h}_n)] \leq 0.5.$$

Thus this task is not weakly OOD learnable despite the positive transfer. ■

Note that in the following results, meta-learning refers to scenarios in which we have target sample size 0, i.e., zero data sampled from the evaluation distribution of a given task.

Theorem 4.4 (Weak OOD "Meta" Learner Theorem). *There exists tasks t such that with zero data sampled from the evaluation distribution of that task, f can weakly OOD learn, but f does not strongly OOD learn.*

Proof. Consider the following scenario: The query space is the unit interval, $\mathcal{X} = [0, 1]$. Each point $x \in \mathcal{X}$, is assigned a binary label (the action space $\mathcal{Y}$ is $\{0, 1\}$). In other words, this is a simple realizable 0-1 classification task with zero Bayes error, $R^* = 0$ (that is, there exists classifiers with no errors). Let the risk be the expected loss, where the loss is 0-1 loss. Let the hypothesis space be all functions $\mathcal{H} = \{h | h : [0, 1] \to \{0, 1\}\}$, and the learners be any learner. Consider distributions of the form

$$P(Y = y|X = x) = \mathbb{I}(x \leq z)\mathbb{I}(y = 0) + \mathbb{I}(x > z)\mathbb{I}(y = 1),$$

where $\mathbb{I}(a)$ is the indicator function that condition a is true, $z \in [0, 1]$ and $X \sim U([0, 1])$. Assume that the statistical model $\mathcal{P}_{X,Y}$ is the set of distributions where $z \in [0.7, 1]$. We will use P_{z_0} to mean a distribution in $\mathcal{P}_{X,Y}$ with $z = z_0$. Assume we observe $n = 0$ samples from the true but unknown $P_z \in \mathcal{P}_{X,Y}$. Further assume that the data pairs drawn according to $P_{\mathbf{S}}$ with the same form as distributions in $\mathcal{P}_{X,Y}$ but with $z = 0.6$. That is, $P_{0.6} \notin \mathcal{P}_{X,Y}$, and thus we are in an OOD learning setting.

This problem is weakly OOD learnable. For example, the learner that takes the biggest x_0 with label 0 in the data and outputs hypotheses that labels any $x > x_0$ as 1 and 0 otherwise will perform better than chance with any desired probability $1 - \delta$ given enough data for any $P_{X,Y} \in \mathcal{P}_{X,Y}$. This setting is not strongly OOD learnable however. Assume for the sake of contradiction that we have some strong learner f for this setting. Consider $P_{0.7} \in \mathcal{P}_{X,Y}$. If we let $\delta = \epsilon = 0.01$, then because f is a strong OOD learner, there exists some N such that with probability $1 - \delta = 0.99$, the learner outputs hypotheses whose risk only differs from the optimal risk by $\epsilon = 0.01$ whenever $m \geq M$. The risk here is calculated as

$$R_{P_{0.7}}(\hat{h}_{0,m}) = \int_0^1 \mathbb{I}(\hat{h}_{0,m}(x) \neq h^*_{0.7}(x))dx,$$

where $h^*_{0.7}$, which outputs 0 if $x \leq 0.7$, and 1 otherwise, is the optimal hypothesis. $R_{P_{0.7}}(h^*_{0.7}) = 0$ in this scenario. Thus, by our assumption, $R_{P_{0.7}}(\hat{h}_{0,m}) < 0.01$ with probability greater than 0.99. This implies that f outputs hypotheses which differ from $h^*_{0.7}$ on at most 1% of $[0, 1]$.

Now, take $P_{0.9} \in \mathcal{P}_{X,Y}$ as the target distribution and let $h^*_{0.9}$ be the optimal hypothesis for this distribution. Since the source data does not change, f will output the same hypotheses as above for $n \geq N$. These hypotheses will differ from $h^*_{0.9}$ by more than 1% at least 99% of the time no matter how much out-of-task data we get (because they differ by at most 1% from $h^*_{0.7}$). In other words, at least 99% of the time, $R_{P_{0.9}}(\hat{h}_n) \geq \epsilon = 0.01$ no matter how high we make m. This contradicts f being a strong OOD learner for the given setting, and so it is not strongly OOD learnable. That is, the setting is weakly OOD learnable but not strongly OOD learnable, which is the desired result. ■

An immediate implication of the above theorem is that the class of weakly OOD learning problems is smaller than the class of strongly OOD learning problems.

Proposition 4.5 (Weak meta-learning does not imply non-uniform meta-learning). *There are tasks t for which weak OOD meta-learning does not imply non-uniform OOD meta-learning.*

Proof. The exact same proof as for theorem 4.4 implies this result. ∎

5 Formalizing distinct learning paradigms In this section we apply our notion of learning efficiency to characterize whether, and the extent to which, a learner has learned in various different learning scenarios. We made a number of choices based on the literature to formalize these learning scenarios, some of which are not universally agreed upon. That said, the names of the different learning scenarios are only important for clarity of communication; our main intent with this section is to illustrate the flexibility of our proposed formalism.

5.1 In-distribution learning In-distribution learning is the simplest learning scenario. It includes as special cases hypothesis testing, point estimation, unsupervised learning, supervised learning, federated learning, online learning, forecasting, games, and reinforcement learning.

Let h_0 be the hypothesis output by the learner prior to seeing $\mathbf{S}_n$. In the *tabula rasa* learning mentality (that is, where learning happens upon a blank slate [47]), h_0 would be the hypothesis prior to seeing *any* data. However, more generally, it is simply the hypothesis prior to seeing a new dataset, $\mathbf{S}_n$. We define $\mathbf{S}_0$ to be the 'dataset' prior to seeing $\mathbf{S}_n$, which might be the empty dataset, and we let $f(\mathbf{S}_0) = h_0$ denote the 'base hypothesis'. $\mathbf{S}_0$ can be quite general; specifically, $\mathbf{S}_0$ could be a hypothesis learned from a previous dataset, or a prior obtained from another dataset, etc. Recall that our definition of the data sample space $\mathcal{S}^*$ above was unrestricted. h_0 is therefore a function of the inductive bias of the learner f, as well as its hyper-parameters, and 'initial conditions' (such as the randomization of the weights prior to training a deep network), as well as anything else learned from $\mathbf{S}_0$. Note that h_0 may, in general, also be a random variable, though we ignore that source of variability here. h_0 could also be the naïve chance hypothesis, which assumes a non-informative prior on the action space. From a Bayesian perspective, $\hat{h}_0$ corresponds to our prior knowledge for how to respond to a given query, and $\hat{h}_n$ corresponds to our posterior updated knowledge, based on dataset $\mathbf{S}_n$.

Application 5.1 (Learning Efficiency for In-Distribution Learning). *The **learning efficiency** of an in-distribution learner f for task t given datasets $\mathbf{S}_n$ is the learning efficiency for task t with a $\mathbf{S}^A = \mathbf{S}_0$ and $\mathbf{S}^B = \mathbf{S}_n$,*

$$\mathsf{LE}_f^t(\mathbf{S}_0, \mathbf{S}_n) = \frac{\mathcal{E}_f^t(\mathbf{S}_0)}{\mathcal{E}_f^t(\mathbf{S}_n)} \quad .$$

We say that f learns in this task if $\log \mathsf{LE}_f^t > 0$.

This definition has conditioned on a particular sample size, n. One could generalize this definition to be a function of n, and define the learning efficiency curve for a given learner f on this task.

5.2 Transfer Learning Transfer learning is arguably the simplest OOD learning scenario with one underlying task t along with an explicit set of J datasets $\{\mathbf{S}^1, \ldots, \mathbf{S}^J\}$ [48, 49]. $\mathbf{S}^1$ here is data associated with task t, often called the *target data*; that is, it is assumed to be drawn according to the evaluation distribution for task t. Without loss of generality, we assume that $\mathbf{S}^1$ includes the base dataset, $\mathbf{S}_0$, that is, $\mathbf{S}_0 \subset \mathbf{S}^1$ (thus, $|\mathbf{S}^1| \geq 1$, because it always includes $\mathbf{S}_0$ at least). The other datasets, $\mathbf{S}^2, \ldots, \mathbf{S}^J$—sometimes called side information [40], or 'source data'—are available to assist in learning and producing

a better hypothesis. Let $\mathbf{S}_n = \bigcup_{j=0}^{J} \mathbf{S}^j$ be the amalgamated dataset. While the problem associated with this task is to produce a hypothesis that minimizes the risk for task t; the fundamental question in transfer learning is whether a learner has successfully leveraged the out-of-distribution (or source) data to improve performance. That is, has the *generalization error decreased* by virtue of leveraging $\mathbf{S}^2, \ldots, \mathbf{S}^J$, relative to only using $\mathbf{S}^1$?

To answer such a question, we consider the performance $\mathcal{E}_f^t(\mathbf{S}^1)$ of the hypothesis learned by f using only task t data, as compared to the hypothesis learned by f using all the data $\mathcal{E}_f^t(\mathbf{S}_n)$.

Application 5.2 (Learning Efficiency for Transfer Learning). *The **transfer learning efficiency** of a transfer learner f for task t given datasets $\mathbf{S} = \{\mathbf{S}^1, \ldots, \mathbf{S}^J\}$ is the learning efficiency for task t with a $\mathbf{S}^A = \mathbf{S}^1$ and $\mathbf{S}^B = \mathbf{S}_n$,*

$$\mathsf{LE}_f^t(\mathbf{S}^1, \mathbf{S}_n) = \frac{\mathcal{E}_f^t(\mathbf{S}^1)}{\mathcal{E}_f^t(\mathbf{S}_n)} \ .$$

We say that f transfer learns in this supertask if $\log \mathsf{LE}_f^t > 0$.

The above definition is identical to the definition of learning efficiency, with $\mathbf{S}^A$ and $\mathbf{S}^B$ denoting different datasets. In other words, transfer learning is a generalization of in-distribution learning (when $\mathbf{S}^1 = \mathbf{S}_0$). Stated differently, in-distribution learning is a special case of transfer learning, but in-distribution learning transfers from priors and inductive biases rather than source data.

While others often include a source task in defining transfer learning, in our definition, we only specify one task, that is, the target task. Source data may have a source task associated with it, but from the perspective of optimizing learning efficiency for the target task, there is no need to specify a source task with its associated risk, etc. Note that we have conditioned on the sample size for the target dataset, as well as the source datasets. Doing so enables us to avoid making any assumptions about the total number of possible datasets, or the distribution over those datasets. One could define a notion of learning efficiency with respect to a distribution over datasets and sample sizes, and take additional expectations to obtain something like expected learning efficiency.

5.3 Multitask Learning Multitask learning can be thought of as a generalization of transfer learning [50, 51]. Assume the existence of an **environment** of tasks $\mathcal{T}$, with cardinality $|\mathcal{T}|$, along with J datasets $\mathcal{J} = \{\mathbf{S}^1, \ldots, \mathbf{S}^J\}$. We let $\mathbf{S}_n$ be all of the data (i.e. the amalgamated dataset as defined above, including the base data). For simplicity of presentation, and without loss of generality, we assume that all of the tasks share a common data space, query space, action space, etc.; that is, the tasks share a common setting. The tasks can differ on the datasets that come with them, their distributions, risks, and errors. Let $\mathbf{S}^t$ be the data associated with task $t \in \mathcal{T}$ drawn according to the data distribution of task t. Note that it is possible that $J \geq |\mathcal{T}|$. Of course, the dataset $\mathbf{S}_t$ associated with a given task t may have no data, in which case we have no target data for that task. Any dataset $\mathbf{S}^t$ with $t > |\mathcal{T}|$ is not associated with any specific task. It could be the case that a given dataset is associated with multiple tasks; in that case, we simply 'copy' the dataset as many times as necessary to ensure there that each task has one dataset associated with it. Just like in transfer learning, we want to measure whether performance improves on each task by virtue of seeing out-of-distribution data. Let $\mathcal{E}_f^t(\mathbf{S}^t)$ denote the generalization error of the hypothesis learned by f on task t using only the data associated with that task, $\mathbf{S}^t$. For notational simplicity, we assume each of $\mathbf{S}^t$ includes the base data $\mathbf{S}_0$, that is, $\mathbf{S}_0 \subset \mathbf{S}^t$. And let $\mathcal{E}_f^t(\mathbf{S}_n)$ denote the generalization error of the hypothesis learned by f on task t using all available data, $\mathbf{S}_n$. Now, we can define multitask learning efficiency.

Application 5.3 (Learning Efficiency for Multitask Learning). *The **multitask learning efficiency** of f for tasks $\mathcal{T}$ given datasets $\mathcal{J}$ is the set of learning efficiencies with the task t risks, with $\mathbf{S}^A = \mathbf{S}^t$, and $\mathbf{S}^B = \mathbf{S}_n$,*

$$\mathsf{LE}_f^t(\mathbf{S}^t, \mathbf{S}_n) = \frac{\mathcal{E}_f^t(\mathbf{S}^t)}{\mathcal{E}_f^t(\mathbf{S}_n)}, \qquad t \in \mathcal{T}.$$

In other words, in multitask learning, we have a set of $|\mathcal{T}|$ learning efficiencies, one for each of the tasks.

Note that this is another instance of learning efficiency, with $\mathbf{S}^A$ and $\mathbf{S}^B$ denoting different datasets. The multitask setting naturally leads to questions about performance *across* tasks. To quantify learning across tasks we introduce a non-negative weight per task, $\boldsymbol{w} = (w_1, w_2, \ldots)$, where $\sum_{t\in\mathcal{T}} w_t = 1$.

Application 5.4 (Multitask Learning). *Learner f $\boldsymbol{w}$-multitask learns if the log of the convex combination of learning efficiencies of each task is positive. That is, f multitask learns if*

$$\log \sum_{t\in\mathcal{T}} w_t \cdot \mathsf{LE}_f^t(\mathbf{S}^t, \mathbf{S}_n) > 0 \ .$$

The advantage of having a general set of weights is that it allows a varying measure depending on the situation. For example, we may have multiple tasks but only care about transfer to one task. In that case, all of the weight would be placed on the learning efficiency of that task, and we are back to the case of transfer learning with one task. The other tasks' data can then be thought of as just more buckets of side information. This shows that multitask learning is a generalization of transfer learning. Alternatively, if we let $w_t = 1/|\mathcal{T}|$ for all $t \in \mathcal{T}$, then we have a version of multitask learning that could be considered *weak multitask learning*. In this scenario, we say f has multitask learned whenever the log of the *average* learning efficiency is positive. A learner f strong multitask learns in an environment $\mathcal{T}$ whenever it $\boldsymbol{w}$-multitask learns for any feasible vector of weights $\boldsymbol{w}$. An immediate implication of strong multitask learning is that f positively transferred for *all* tasks in the environment, that is, each task has benefited from the out-of-distribution data. In general, for any set of weights $\boldsymbol{w}$, if a learner multitask learns, it has necessarily transfer learned for at least one of the $|\mathcal{T}|$ tasks.

5.4 Meta-Learning Loosely, meta-learning is learning to learn [52]. The learner has some past experiences, and learns to learn only if these past experiences help it learn in new situations more effectively. In practice, meta-learning is very related to multitask learning. Again, assume we have an environment $\mathcal{T}$, but unlike in typical multi-task learning scenarios the cardinality of the environment, $|\mathcal{T}|$, could be infinite. And further assume that we have already faced K of the tasks, and label the set of already observed tasks $\mathcal{T}_K$. As before, we have J datasets, $\{\boldsymbol{S}^1, \ldots, \boldsymbol{S}^J\}$, where $J \geq |\mathcal{T}_K|$. Now, we face a task, t which may or may not be an element of $\mathcal{T}_K$ and acquire a new dataset, $\boldsymbol{S}^{J+1}$. The question of meta learning is, did learning tasks $\mathcal{T}_K$ using data $\{\boldsymbol{S}^j\}_{j=1}^J$ help us learn more efficiently for task t? Intuitively, we we want to measure if we learn task t more efficiently with $\mathcal{J} = \{\boldsymbol{S}^1, \boldsymbol{S}^2, \ldots, \boldsymbol{S}^{J+1}\}$, as compared to learning task t with just $\boldsymbol{S}^{J+1}$. The new data may arrive with complete task information (i.e., the task aware learning setting), no task information (i.e., the task unaware or agnostic learning setting), or partial information (i.e., the task semi-aware learning setting). As in multitask learning, a weight is associated to each task $\boldsymbol{w} = \{w_1, w_2, \ldots\}$ over all $t \in \mathcal{T}$. We further constrain the weights to form a discrete probability distribution, that is, $\sum_{i=1}^{\infty} w_i = 1$, $w_i \geq 0$. Then,

Application 5.5 (Meta-Learning). *Learner f $\boldsymbol{w}$-meta-learns in environment $\mathcal{T}$ using dataset $\mathcal{J}$ if*

$$\log \sum_{t \in \mathcal{T}} w_t \cdot \mathsf{LE}_f^t(\mathbf{S}^t, \mathbf{S}_n) > 0.$$

The implication of the above definition is that if f has $\boldsymbol{w}$-meta-learned, then, it will perform better on tasks in the environment than it would have if it had not observed the previous K tasks and J datasets. If $\mathcal{T} = \mathcal{T}_K$, then meta-learning is simply multitask learning, indicating that meta-learning is a generalization of multitask learning. The key difference between the two is in meta-learning there are unobserved tasks, whereas in multitask learning, the learner has *a priori* access to all tasks. Weak meta-learning, like weak multitask learning, corresponds to the setting where the log of the *average* learning efficiencies is positive. Strong meta-learning, like strong multitask learning, corresponds to the setting where the log of each learning efficiency is positive, that is, f positively transfers for each task in the environment. Note that sometimes meta-learning is defined in a similar fashion, except that $t \notin \mathcal{T}_K$, such that f learns faster specifically on *new* tasks.

Our definition of meta-learning generalizes Baxter's in a few ways [11]. First, Baxter did not allow for additional source datasets, rather, there was a one-to-one mapping between datasets and tasks. Second, Baxter was only interested in hypotheses that performed well for any distribution in the environment, akin to our strong meta-learning.

5.5 Continual Learning Continual learning generalizes multi-task and meta-learning, and therefore, some variants of it are referred to as online meta-learning [53]. There are two key points of departure. First, in meta-learning, we did not assume a distribution over tasks, rather, just a weight assigned to how much one cares about a task. In continual learning, tasks are governed by a (potentially non-stationary and even adversarial) stochastic process, akin to the random processes governing the data in online learning (see Shalev-Shwartz et al. [26] for more details). Second, unlike meta-learning, but like streaming learning, continual learning necessarily includes computational complexity constraints (which are often not explicitly enumerated) [24], as described below.

Several works have explored this scenario as well as presented novel learners for it [43, 44, 54, 55]. Here we formalize the framework they explored using the above established conventions. In the most general version of continual learning, we have streaming data, queries, actions, error, and *tasks*—in contrast to in-distribution online learning, where the task is fixed *a priori*. We assume that each (data, queries, actions, error, and tasks) is governed by a stochastic process. While in online learning, sequential learning, and reinforcement learning, only the distribution of the data may change, in continual learning, other aspects of the task may also be dynamic. In other words, in other learning scenarios the setting is fixed; in contrast, in continual learning, the setting can dynamically (and even adversarially) change. Hence, anything about a task can change in any moment (although the learner might not be aware of everything that has changed). A particularly interesting and special case of continual learning occurs when the data arrive in batches from the same task.

A second distinction between continual learning learning and some sequential learning scenarios is that continual learning necessarily includes computational complexity constraints [56]. If it did not, then the learner f (or hypothesis h) could store all the data $\mathbf{S}_n$, and retrain everything from scratch. Such a strategy would naïvely require $\mathcal{O}(n)$ space and $\sum_{i=1}^{n} i = \mathcal{O}(n^2)$ time (assuming each data sample is the same size). Therefore, a *bona fide* continual learner must, at a minimum, have a computational upper bound of $o(n)$ space and/or $o(n^2)$ time, and should in practice be far more efficient.

Intriguingly, many previously proposed purported continual learners do not satisfy the above mentioned computational complexity constraints. Vogelstein et al. [24] quantifies the complexity of several

state-of-the-art claimed continual learners, and shows that Elastic Weight Consolidation [25] and Progressive Neural Networks [57] fail to satisfy the computational complexity constraints established above, rendering them not *bona fide* continual learning algorithms (although some generalizations of them both do satisfy these criteria).

As mentioned above, for the formalism in described in §3 to account for continual (and lifelong) learning, it must be generalized somewhat to replace the random variables with stochastic processes, incorporate a stochastic process on tasks, and generalize the class of learners to include updating hypotheses.

Quantifying continual learning New quantities of interest arise as well due to the sequential nature of learning in this environment. For example, how much do past tasks help us in learning for the current task? How much do future tasks help us in learning for the past tasks? These questions lead us to define evaluation criteria (informally introduced in Vogelstein et al. [24]). First, given a task of interest t, we define $\mathbf{S}^{<t}$ to be the set of data points up to and including the last data point from task t.

Application 5.6 (Learning Efficiency for Forward Transfer). *The **forward learning efficiency** of f for task t given n samples is the learning efficiency with the task t risk, $\mathbf{S}^A = \mathbf{S}^t$, and $\mathbf{S}^B = \mathbf{S}^{<t}$,*

$$\mathsf{LE}_f^t(\mathbf{S}^t, \mathbf{S}^{<t}) \tag{5.1}$$

We say that f forwards learns if the forward learning efficiency is greater than 1, or equivalently, if $\log \mathsf{LE}_f^t > 0$.

Application 5.7 (Learning Efficiency for Backward Transfer). *The **backward learning efficiency** of f for task t given n samples is the learning efficiency with the task t risk and $\mathbf{S}^A = \mathbf{S}^{<t}$ and $\mathbf{S}^B = \mathbf{S}_n$,*

$$\mathsf{LE}_f^t(\mathbf{S}^{<t}, \mathbf{S}_n) \tag{5.2}$$

We say that f backward learns if the backward learning efficiency is greater than 1, or equivalently, if $\log \mathsf{LE}_f^t > 0$.

Note that this concept of backward transfer is closely related to "reverse transfer" which was first introduced in Ruvolo and Eaton [13]. Multiplying forward and backward transfer efficiency for a particular task recovers learning efficiency for that task:

$$\mathsf{LE}_f^t(\mathbf{S}^t, \mathbf{S}) = \mathsf{LE}_f^t(\mathbf{S}^t, \mathbf{S}^{<t}) \times \mathsf{LE}_f^t(\mathbf{S}^{<t}, \mathbf{S}) \ .$$

Putting all the above together, we can now define whether f has continually learned. As before with meta-learning, we assume that the weights form a discrete probability distribution.

Application 5.8 (Continual Learning). *Learner f $\boldsymbol{w}$-continually learns tasks $t \in \mathcal{T}$ if the log of the convex combination of learning efficiencies is greater than 0. That is, f $\boldsymbol{w}$-continually learns if*

$$\log \sum_{t \in \mathcal{T}} w_t \cdot \mathsf{LE}_f^t(\mathbf{S}^t, \mathbf{S}) > 0 \ .$$

Weak and strong continual learning are defined as in weak and strong multitask and meta-learning. If one sets the weights of all the unobserved tasks to zero, then continual learning above corresponds to continual *multitask* learning. On the other hand, allowing the weights of unobserved tasks to be positive, then the above corresponds to continual *meta*-learning. It may sometimes be desirable to assert that f continually learns only if it achieves positive forward and backward $\boldsymbol{w}$-learning efficiency.

5.6 Lifelong Learning While others have used the terms continual and lifelong learning interchangeably, here, we argue that there can be important meaningful differences between the two. Specifically, we advocate that lifelong learning is a generalization of continual learning that renders it closer to biological learning. The key difference between lifelong learning and continual learning is that in lifelong learning, two things can change over the lifetime of the agent. First, the weights on each task may vary over time. In continual learning, the weights are typically specified *a priori*, and are fixed over the duration of learning. In contrast, in lifelong learning, the value of any individual task may change over time. For example, frogs do not need to know how to swim as tadpoles after they metamorphose, so the weight on swimming as a tadpole goes down to zero at that point. Second, the computational constraints can change over the lifetime of the learner. Specifically, learners essentially go through three life phases. First, while the learner is juvenile, it can increase its resources as more data and tasks are presented to it. In this phase, it builds representational capacities. Second, while the learner is an adult, it has fixed resources, and therefore, must leverage and recombine existing resources to obtain new capabilities, but cannot grow new resources. Third, while the learner is a geriatric, it has diminishing resources, and therefore, must re-prioritize its allocation of resources to ensure that it does not forget crucial skills (such as how to breath). These resource constraints can be expressed in terms of the sets of hypotheses and learners under consideration at each time. To our knowledge, there are no existing algorithms that currently can satisfy these additional complications, despite the fact that all biological learning agents must address them. Let $w_{t,u}$ correspond to the weight on task t at time u, so $\boldsymbol{w} = (w_{t,u})$, and let $\mathsf{LE}_f^{t,u}$ denote the learning efficiency for task t that may be changing with u.

Application 5.9 (Lifelong Learning). *Learner f $\boldsymbol{w}$-lifelong learns tasks $t \in \mathcal{T}$ if the log of the convex combination of learning efficiencies is greater than 0. That is, f $\boldsymbol{w}$-lifelong learns if*

$$\log \sum_{u>0} \sum_{t \in \mathcal{T}} w_{t,u} \cdot \mathsf{LE}_f^{t,u}(\mathbf{S}^t, \mathbf{S}) > 0 \ .$$

6 Discussion

6.1 A unified quantification of learning The key quantity we introduced in this work is learning efficiency (§2). The formalization of this quantity enables us to convert questions of whether we have learned, transfer learned, multitask learned, etc. into questions about whether the learning efficiency is log positive. Learning efficiency also provides a measure to quantify the amount a given learner has learned. Given an appropriate learner f, and the choice of datasets $\mathbf{S}_A$ and $\mathbf{S}_B$, learning efficiency can quantify the extent of learning in many in-distribution and out-of-distribution learning tasks. More specifically, $\mathsf{LE}_f^t(\mathbf{S}_A, \mathbf{S}_B)$ can correspond to the following different learning scenarios:

- In-distribution learning: $\mathbf{S}^A = \mathbf{S}_0$ and $\mathbf{S}^B = \mathbf{S}_n$.
- Transfer learning: $\mathbf{S}^A = \mathbf{S}^1$ and $\mathbf{S}^B = \mathbf{S}_n$.
- Multitask learning for task t: $\mathbf{S}^A = \mathbf{S}^t$ and $\mathbf{S}^B = \mathbf{S}_n$.
- Forward Learning for task t: $\mathbf{S}^A = \mathbf{S}^t$ and $\mathbf{S}^B = \mathbf{S}^{<t}$.
- Backward Learning for task t: $\mathbf{S}^A = \mathbf{S}^{<t}$ and $\mathbf{S}^B = \mathbf{S}_n$.

In all of the above cases, we have (transfer, multitask, etc.) learned if the log of learning efficiency is positive. Multitask, meta, continual, and lifelong learning can then be defined as linear combinations of learning efficiencies for each of the tasks. This equivalence shows that all of the above-described learning scenarios can reasonably be thought of as different special cases of out-of-distribution learning; the difference between the different learning paradigms then comes down to what one is transferring to and from.

This framework allows us to compare the performance of different learners on a task via their *learning curves*, which is the sequence of learning efficiency as a function of sample size. In other words, a learning curve for task t is the function, $(\mathrm{LE}_f^t(\mathbf{S}_0, \mathbf{S}_n))$, as $n = 0, 1, 2, \ldots$. The learning curve of a particular learner f in a particular task t completely characterizes how well it learns a task t as a function of out-of-distribution sample size, including zero-shot and few-shot learning scenarios. Differences in large sample learning rates, and asymptotics, are therefore summary statistics of this function.

The above formal definition of whether f has learned dovetails nicely with Mitchell's definition and weak OOD learnability. Specifically, f learns about task t from data $\mathbf{S}^B \backslash \mathbf{S}^A$ whenever its error decreases specifically due to the additional data. It need not decrease by any specific amount, and it need not have a sufficiently large sample size.

6.2 A high-dimensional landscape of kinds of learning problems A number of previous papers have endeavored to developed taxonomies of transfer and/or continual learning [49, 58–61]. However, a consequence of the generalized learning task defined here, any aspect of the setting for each task may change, each of these taxonomies is a low-dimensional projection of a high-dimensional landscape of potential continual learning problems. We therefore propose a unifying hierarchy (Figure 1), ranging from the simplest in-distribution learning tasks to the most complex biological learning tasks. Note that even the lowest levels of the hierarchy contains many disparate kinds of tasks. For example, in-distribution learning contains as special cases essentially all of classical machine learning, including point estimation, hypothesis testing, supervised learning, unsupervised learning, casual learning, federated learning, streaming learning, online learning, and reinforcement learning. Even with this list of different in-distribution learning problems, one can partition the space of problems in multiple ways: batch vs. online, supervised vs. unsupervised vs. reinforcement learning, perceptual vs. action, independent data samples or not, etc.

Once we are faced with multiple datasets, as in transfer learning, already the landscape gets incredibly more complex. For example, given a pair of datasets, how are the query spaces related? Are they the same space, overlapping spaces, one subspace is a strict subset of the other, or non-overlapping spaces? The same question can be applied to the action space. The statistical model (the set of admissible distributions) has similar questions, for example, if one dataset sampled from a mixture distribution, where one of the components corresponds to the distribution of the other dataset, are the distributions related by a rigid, linear, affine, or nonlinear transformation, etc.? Adding multiple tasks, as in multitask and meta-learning further complicates things. For example, is it clear to the learner and/or the hypothesis which task each query is associated with? how much information about each tasks' setting is provided for any given query? Which components of the settings associated with each task differ from one another: (1) query space, (2) action space, (3) hypothesis space, (4) risk, (5) distribution, (6) statistical model, (7) evaluation distribution, (8) learner space, or (9) error? Adding dynamics, as in continual and lifelong learning further exacerbates these issues. For example, do the computational space and/or time constraints change for the learner and/or hypothesis? If so, in which ways? Similarly, for some tasks data could arrive in batches, in others it could arrive sequentially, and the same is true for tasks.

The consequence of this inherent flexibility in defining generalized learning tasks complicates the literature. Any given paper on 'continual learning' could be solving one of many different kinds of problems. Assume for the moment that a given paper is addressing a set of supervised learning classification tasks. Given the nine different components above, and assuming only two different choices for each component, yields $2^9 = 512$ total possible continual learning classification problems, and different approaches will be designed typically to only address a very small subset of them. We therefore recommend greater

specificity in manuscripts to clarify precisely the scope of the proposed learner and/or theory.

Another consequence of this formalism is that it exposes that many previously proposed continual and lifelong learning algorithms are, in fact, not respecting the computational complexity constraints to render them *bona fide* continual learning. Even for those that are, comparing algorithms with different computational complexity bounds is a bit like comparing apples to oranges. So, we advocate for more explicit theoretical and empirical investigations of computational complexity of these algorithms to understand their relative trade-offs.

6.3 Conjectures for future work

Conjecture 6.1 (Non-uniform does not imply weak). *There are tasks t for which non-uniform OOD learning does not imply weak OOD learning.*

If we had task data, but not too much, how well can we perform with lots of out-of-task data? We conjecture that one cannot perform arbitrarily well, and leave the proof for future work.

Conjecture 6.2 (Weak OOD Learning Theorem). *Assume that learners cannot be constant on data. Let $\epsilon, \delta > 0$. Assume that to meet the strong OOD learning bound with both task data and out-of-task data, we need N task distribution data points and 0 out-of-task data points. Then there exists scenarios in which if $n < N$ (number of task data points), then no matter how many out-of-task data points we have, we cannot meet the strong OOD learning bound, but we can weakly OOD learn.*

Upper bounds on learning efficiency One of course naturally wonders, how high can we make the learning efficiency? That is, given that $\mathbf{S}^A$ is drawn according to some distribution, then for a given number of data points in $\mathbf{S}^B$, how high can the learning efficiency get? The only thing we can control here is the distribution of $\mathbf{S}^B$. Under certain general assumptions, the best possible distribution according to which to draw $\mathbf{S}^B$ is the evaluation distribution. That is, the distribution according to which the expected risk is calculated.

6.4 Limitations of the framework Our proposed framework attempts to unify in-distribution and out-of-distribution learning definitions; though it has several limitations. Perhaps most importantly from a machine learning perspective, we have not provided any theorems stating *when* a given learner can solve a particular out-of-distribution learning task. While of the utmost importance to establish the theoretical utility of this framework, we leave it to future work. One of our motivating goals was to establish a learning framework that was sufficiently general to characterize both biological and machine learning. However, this framework is inadequate for characterizing biological learning a few reasons.

First, in biology, there are generally no explicit tasks. Other forms of lifelong learning (e.g., Sutton et al. [62]) operate in a task-free setting, viewing the concept of a "task" as a convenient yet unrealistic construct. For example, what task are you doing right now? Are you reading this paper, sitting balanced in your chair, listening to your surroundings, classifying typeset characters into letters and then words, recongizing speling errors, or some combination thereof? What precisely constitutes a task? Sutton et al. [62] operates by continually learning a set of functions (specifically, generalized value functions) that predict different aspects of the world and can be combined together hierarchically to achieve objectives. These functions are learned from a continual sensorimotor stream, without any extrinsic notion of tasks. To map such a task-free lifelong learner into our framework, we could view each of these prediction functions as a different intrinsic task that the agent must learn. This permits the machinery we developed to equally apply to this scenario, with the important caveat that the tasks learned by the agent are not specified externally or *a priori*.

Second, there are no discrete 'samples' in biology; rather, biological agents are hit with a lifelong

onslaught of data streams without a synchronizing clock. At any given time a biological agent may be acquiring data, acting, and learning, or nothing at all. Third, for the most part, in biology, there is not simply one kind of unidimensional error. Instead, different kinds of inputs provide different kinds of affordances, such as oxygen, calories, and sleep [63]. In future work, we hope to further bridge the gap between lifelong and biological learning by addressing these three limitations.

6.5 The quest for artificial general intelligence It is unclear how problematic the differences are between our notion of lifelong learning and biological learning. Many have recently argued that modern machine learning and artificial intelligence is hitting a wall [64], or entering a new winter [65]. Various experts in machine learning and computer science have proposed that the main bottlenecks to overcome include causal learning [16]. Other experts that lean more towards cognitive science have proposed that the key bottleneck is symbolic reasoning [66] or the barrier of meaning [67]. And still others believe that if we keep building larger and larger deep networks, with more data and bigger computers (and a few unspecified conceptual breakthroughs) we will eventually close the gap [68–70]. Which of these beliefs will win in the end, if any, remains to be seen, though the fight continues on [71]. We expect that there will be many small victories from a wide diversity of individuals and approaches, that together will enable us to bridge the gap, including but not limited to bigger computers and datasets.

6.6 Concluding thoughts We were motivated to write this manuscript primarily for two reasons. First, as we read the literature, we often found papers with proposed solutions to problems, but it was often unclear to us what problem was actually being solved, and how those problems are related to one another. By virtue of standardizing an approach to quantifying out-of-distribution learning—via learning efficiency—we hope to be able to better understand the current state-of-the-art, and also advance beyond it. Second, one particular way in which we hope to advance beyond the current state-of-the-art is by eclipsing biological learning in additional domains [72]. We believe that a step towards realizing this dream includes establishing a formalism that is sufficiently flexible to be able to coherently evaluate learners in many different learning paradigms, including both biological and machine learners. Indeed, arguably the crux of the gap between artificial general intelligence and natural intelligence is that natural biological learners are able to perform out-of-distribution learning well both within and across lifetimes. We hope our proposed formalism can help clarify both where we are, and also where to go from here.

Acknowledgements The authors thank the support of the National Science Foundation-Simons Research Collaborations on the Mathematical and Scientific Foundations of Deep Learning (MoDL, NSF grant 2031985). This work is graciously supported by the Defense Advanced Research Projects Agency (DARPA) Lifelong Learning Machines program through contracts FA8650-18-2-7834, HR0011-18-2-0025, and FA8750-18-C-0103. Research was partially supported by funding from Microsoft Research and the Kavli Neuroscience Discovery Institute. The authors are grateful for critical feedback from Yoshua Bengio, Anirudh Goyal, Pratik Chaudhari, Rahul Ramesh, Raman Arora, Rene Vidal, Jeremias Sulam, Adam Charles, Anirudh Goyal, Timothy Verstynen, Konrad Kording, Jeffrey Dick, Erik Peterson, Weiwei Yang, Chris White, and Iris van Rooij.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems, pages 1097–1105, 2012.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. October 2018.

[3] Scott Mayer McKinney, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back, Mary Chesus, Greg C Corrado, Ara Darzi, Mozziyar Etemadi, Florencia Garcia-Vicente, Fiona J Gilbert, Mark Halling-Brown, Demis Hassabis, Sunny Jansen, Alan Karthikesalingam, Christopher J Kelly, Dominic King, Joseph R Ledsam, David Melnick, Hormuz Mostofi, Lily Peng, Joshua Jay Reicher, Bernardino Romera-Paredes, Richard Sidebottom, Mustafa Suleyman, Daniel Tse, Kenneth C Young, Jeffrey De Fauw, and Shravya Shetty. International evaluation of an AI system for breast cancer screening. Nature, 577(7788):89–94, January 2020.

[4] Nick Statt. OpenAI's Dota 2 AI steamrolls world champion e-sports team with back-to-back victories. https://bit.ly/3o0ZLHs, April 2019. Accessed: 2020-1-25.

[5] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander W R Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Improved protein structure prediction using potentials from deep learning. Nature, January 2020.

[6] Terrence J Sejnowski. The unreasonable effectiveness of deep learning in artificial intelligence. Proc. Natl. Acad. Sci. U. S. A., 117(48):30033–30038, December 2020.

[7] Yoshua Bengio, Frédéric Bastien, Arnaud Bergeron, Nicolas Boulanger-Lewandowski, Thomas Breuel, Youssouf Chherawala, Moustapha Cisse, Myriam Côté, Dumitru Erhan, Jeremy Eustache, Xavier Glorot, Xavier Muller, Sylvain Pannetier Lebeuf, Razvan Pascanu, Salah Rifai, François Savard, and Guillaume Sicard. Deep Learners Benefit More from Out-of-Distribution Examples. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, volume 15 of Proceedings of Machine Learning Research, pages 164–172, Fort Lauderdale, FL, USA, 2011. JMLR Workshop and Conference Proceedings.

[8] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. February 2018.

[9] Michael A Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4845–4854. openaccess.thecvf.com, 2019.

[10] Sebastian Thrun. Is learning the n-th thing any easier than learning the first? In D S Touretzky, M C Mozer, and M E Hasselmo, editors, Advances in Neural Information Processing Systems 8, pages 640–646. MIT Press, 1996. URL http://papers.nips.cc/paper/1034-is-learning-the-n-th-thing-any-easier-than-learning-the-first.pdf.

[11] Jonathan Baxter. A Model of Inductive Bias Learning. J. Artif. Intell. Res., 12(1):149–198, March 2000. URL http://dl.acm.org/citation.cfm?id=1622248.1622254.

[12] Martin Arjovsky. Out of Distribution Generalization in Machine Learning. March 2021.

[13] Paul Ruvolo and Eric Eaton. ELLA: An Efficient Lifelong Learning Algorithm. In International Conference on Machine Learning, volume 28, pages 507–515, February 2013. URL http://proceedings.mlr.press/v28/ruvolo13.html.

[14] David Lopez-Paz and Marc' Aurelio Ranzato. Gradient Episodic Memory for Continual Learning. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, Advances in Neural Information Processing Systems 30, pages 6467–6476. Curran Associates, Inc., 2017.

[15] Natalia Díaz-Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. Don't forget, there

is more than forgetting: new metrics for Continual Learning. October 2018.
[16] Judea Pearl. The seven tools of causal inference, with reflections on machine learning. Commun. ACM, February 2019.
[17] Haitham Bou Ammar, Eric Eaton, Jose Marcio Luna, and Paul Ruvolo. Autonomous cross-domain knowledge transfer in lifelong policy gradient reinforcement learning. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-15), July 2015.
[18] Seungwon Lee, James Stokes, and Eric Eaton. Learning shared knowledge for deep lifelong learning using deconvolutional networks. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), pages 2837–2844, 7 2019.
[19] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain Generalization via Invariant Feature Representation. In International Conference on Machine Learning, volume 28, pages 10–18, February 2013.
[20] H Daumé, III. Frustratingly easy domain adaptation. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics. Citeseer, 2007.
[21] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate Shift by Kernel Mean Matching. In Dataset Shift in Machine Learning. The MIT Press, 2008.
[22] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 1126–1135, International Convention Centre, Sydney, Australia, 2017. PMLR.
[23] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual Learning Through Synaptic Intelligence. of the 34th International Conference on . . . , March 2017.
[24] Joshua T Vogelstein, Jayanta Dey, Hayden S Helm, Will LeVine, Ronak D Mehta, Ali Geisa, Gido M van de Ven, Emily Chang, Chenyu Gao, Weiwei Yang, Bryan Tower, Jonathan Larson, Christopher M White, and Carey E Priebe. Omnidirectional Transfer for Quasilinear Lifelong Learning. April 2020.
[25] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences, 114(13):3521–3526, 2017.
[26] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. J. Mach. Learn. Res., 11:2635–2670, December 2010. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1756006.1953019.
[27] Peter Bickel J and Kjell Doksum A. Mathematical Statistics Basic Ideas and Selected Topics, Volume I, Second Edition. Texts in Statistical Science. Chapman and Hall/CRC, CRC Press Taylor & Francis Group 6000 Broken Sound Parkway NW, Suite 300 Boca Raton, FL 33487-2742, apr 2015.
[28] L G Valiant. A Theory of the Learnable. Commun. ACM, 27(11):1134–1142, November 1984. URL http://doi.acm.org/10.1145/1968.1972.
[29] Shai Shalev-Shwartz and Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, 1 edition edition, May 2014.
[30] Thomas M. Mitchell. Machine Learning. McGraw-Hill, Inc., USA, 1 edition, 1997. ISBN 0070428077.
[31] Gary Marcus and Ernest Davis. Rebooting AI: Building Artificial Intelligence We Can Trust. Pantheon, September 2019.
[32] M Giurfa, S Zhang, A Jenett, R Menzel, and M V Srinivasan. The concepts of 'sameness' and

'difference' in an insect. Nature, 410(6831):930–933, April 2001.
[33] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. Behav. Brain Sci., 40:e253, January 2017.
[34] Anirudh Goyal and Yoshua Bengio. Inductive Biases for Deep Learning of Higher-Level Cognition. November 2020.
[35] V Glivenko. Sulla determinazione empirica delle leggi di probabilita. Gion. Ist. Ital. Attauri., 4:92–99, 1933.
[36] Francesco Paolo Cantelli. Sulla determinazione empirica delle leggi di probabilita. Giorn. Ist. Ital. Attuari, 4(421-424), 1933.
[37] V Vapnik and A Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. Theory Probab. Appl., 16(2):264–280, January 1971.
[38] Ernest Fokoué. Model selection for optimal prediction in statistical machine learning. Not. Am. Math. Soc., 67(02):1, February 2020.
[39] Vladimir Naumovich Vapnik. Statistical learning theory, volume 2 of Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications and Control. Wiley, September 1998.
[40] C E Shannon. Channels with Side Information at the Transmitter. IBM J. Res. Dev., 2(4):289–293, October 1958.
[41] Michael J Kearns and Umesh V Vazirani. An introduction to computational learning theory. The MIT Press. MIT Press, London, England, August 1994.
[42] Robert E Schapire. The strength of weak learnability. Mach. Learn., 5(2):197–227, 1990.
[43] Zeno, Golan, Hoffer, and Soury. Task Agnostic Continual Learning Using Online Variational Bayes. On arXiv, February 2019. URL https://arxiv.org/pdf/1803.10123.pdf.
[44] He, Sygnowski, Galashov, Rusu, Teh, and Pascanu. Task Agnostic Continual Learning via Meta Learning. In 4th Lifelong Learning Workshop at ICML, 2020. URL https://openreview.net/attachment?id=AeIzVxdJgeb&name=pdf.
[45] Pat Langley. Open-World Learning for Radically Autonomous Agents. AAAI, 34(09):13539–13543, April 2020.
[46] Michael McCloskey and Neal J Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In Gordon H Bower, editor, Psychology of Learning and Motivation, volume 24, pages 109–165. Academic Press, January 1989.
[47] Steven Pinker. The Blank Slate: The Modern Denial of Human Nature. Penguin Books, reprint edition edition, August 2003.
[48] L Y Pratt. Discriminability-Based Transfer between Neural Networks. In S J Hanson, J D Cowan, and C L Giles, editors, Advances in Neural Information Processing Systems 5, pages 204–211. Morgan-Kaufmann, 1993.
[49] S J Pan and Q Yang. A survey on transfer learning. IEEE Trans. Knowl. Data Eng., 22(10): 1345–1359, 2010. URL https://ieeexplore.ieee.org/iel5/69/4358933/05288526.pdf.
[50] Samy Bengio, Yoshua Bengio, Jocelyn Cloutier, and Jan Gecsei. On the optimization of a synaptic learning rule. In Preprints Conf. Optimality in Artificial and Biological Neural Networks, volume 2. researchgate.net, 1992.
[51] Rich Caruana. Multitask Learning. Mach. Learn., 28(1):41–75, July 1997. URL https://doi.org/10.1023/A:1007379606734.
[52] Sebastian Thrun and Tom M Mitchell. Lifelong robot learning. Rob. Auton. Syst., 15(1):25–46, July 1995. URL http://www.sciencedirect.com/science/article/pii/092188909500004Y.
[53] Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In

Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 1920–1930, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/finn19a.html.

[54] Tuytelaars Aljundi, Kelchtermans. Task-Free Continual Learning. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/papers/Aljundi_Task-Free_Continual_Learning_CVPR_2019_paper.pdf.

[55] Lee, Ha, Zhang, and Kim. A Neural Dirichlet Process Mixture Model For Task-free Continual Learning. In ICLR 2020, January 2020. URL https://arxiv.org/pdf/2001.00689.pdf.

[56] Iris van Rooij, Mark Blokpoel, Johan Kwisthout, and Todd Wareham. Cognition and Intractability: A Guide to Classical and Parameterized Complexity Analysis. Cambridge University Press, April 2019.

[57] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive Neural Networks. June 2016.

[58] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3712–3722, 2018.

[59] Gido M van de Ven and Andreas S Tolias. Three continual learning scenarios and a case for generative replay. September 2018.

[60] Zhiyuan Chen and Bing Liu. Lifelong Machine Learning, Second Edition. Synthesis Lectures on Artificial Intelligence and Machine Learning, 12(3):1–207, August 2018.

[61] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. September 2019.

[62] Richard S. Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M. Pilarski, and Adam White. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), pages 761–768, 2011.

[63] James J Gibson. The Ecological Approach to Visual Perception: Classic Edition (Psychology Press & Routledge Classic Editions). Psychology Press, 1 edition, November 2014.

[64] Will Knight. Facebook's Head of AI Says the Field Will Soon 'Hit the Wall'. Wired, December 2019.

[65] Filip Piekniewski. AI winter is well on its way. https://blog.piekniewski.info/2018/05/28/ai-winter-is-well-on-its-way/, May 2018. Accessed: 2019-11-24.

[66] Gary Marcus. The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence. February 2020.

[67] Melanie Mitchell. On Crashing the Barrier of Meaning in Artificial Intelligence. AI Magazine, 41(2): 86–92, 2020.

[68] Rich Sutton. The Bitter Lesson. http://www.incompleteideas.net/IncIdeas/BitterLesson.html, March 2019. Accessed: 2021-3-16.

[69] Max Welling. Do we still need models or just more data and compute? https://staff.fnwi.uva.nl/m.welling/wp-content/uploads/Model-versus-Data-AI-1.pdf, April 2019. Accessed: 2021-3-16.

[70] Karen Hao. AI pioneer Geoff Hinton: "Deep learning is going to be able to do everything". MIT Technology Review, November 2020.

[71] DEBATE : Yoshua Bengio | Gary Marcus. https://montrealartificialintelligence.com/aidebate/. Accessed: 2021-3-14.

[72] Ragnar Fjelland. Why general artificial intelligence will not be realized. Humanities and Social Sciences Communications, 7(1):10, June 2020.