Empirically Measuring Transfer Distance for System Design and Operation

Tyler Cody, Stephen Adams, *Associate Member, IEEE*, and Peter A. Beling

***Abstract*—Classical machine learning approaches are sensitive to nonstationarity. Transfer learning can address nonstationarity by sharing knowledge from one system to another, however, in areas like machine prognostics and defense, data are fundamentally limited. Therefore, transfer learning algorithms have little, if any, examples from which to learn. Herein, the authors suggest that these constraints on algorithmic learning can be addressed by systems engineering. We formally define transfer distance in general terms and demonstrate its use in empirically quantifying the transferability of models. We consider the use of transfer distance in the design of machine rebuild procedures to allow for transferable prognostic models. We also consider the use of transfer distance in predicting operational performance in computer vision. Practitioners can use the presented methodology to design and operate systems with consideration for the learning theoretic challenges faced by component learning systems.**

***Index Terms*—Computer vision, prognostics, transfer learning.**

I. Introduction

MACHINE learning is moving from laboratories to the field, however, the identically distributed environments found in the laboratories are rarely found in the real-world. Algorithmic approaches for dealing with nonstationarity rely heavily on data from the new environment, however, such data are not always available.

Applied machine learning for prognostics and health management (PHM) is prototypical of this trend and challenge. Nonstationarities are unavoidable in PHM for machinery. Differences in manufacturing and installment give supposedly identical machines different initial conditions and phenomena, such as degradation, repair, and part replacement cause behavior to drift over a machine's life cycle. Adding to these challenges, labeled data from fielded machines are rarely available because when a failure occurs, the machine is repaired or rebuilt, inducing a distribution change or rendered irreparable.

Similarly, in defense settings, imagery related to new missions is limited. Data collection for new missions is costly. It can require operating in hostile territory or airspace and within enemy field of fire. Moreover, battlefields are dynamic and often do not afford data collection at the scale required by existing data-driven computer vision methods. In addition, defense is game-theoretic in nature and adversaries can manipulate the appearance of concerns, such as aircraft or ground vehicles, to take advantage of an overreliance on data [1].

In both PHM and defense, algorithmic approaches for relating behaviors between systems and over time are fundamentally constrained. Instead of focusing on engineering ever-more adaptive learning systems, the authors suggest a focus on methodologies that support the design and operation of systems to limit nonstationarities to acceptable levels. This interdisciplinary approach treats generalization, i.e., satisfactory predictive performance on new data, as a systems-level goal, not a goal exclusive to algorithm design.

Designing and operating in this way requires metrics that bring the learning theoretic challenges of learning systems to the systems-level. *Transfer distance*, the abstract distance knowledge must traverse to transfer from one system to another, is focal in domain adaptation theory and is used to relate the magnitude of distributional change between systems to prediction error in the new system. Although transfer distance is typically left as an informal notion or implicit in transfer learning methods, here, we formalize it and position it as central to the characterization of the relationship between systems and the generalization of their component learning systems.

We present a Bayesian approach for empirically quantifying transfer distance. The accompanying studies offer a guide for practitioners on how to quantify the difficulty of transfer, the transferability of different learning tasks, and the role of sample size in transferability, as well as how to use transfer distance to quantify expected operational performance. We consider a case in hydraulic actuator health monitoring, where nonstationarities occur as the result of actuator rebuilds. We also consider a case in computer vision with a mission context, where information regarding a mission's expected operating environment is used to assess expected operational performance. We frame the former in terms of system design and the latter in terms of system operation. In doing so, we contribute to the broader effort of developing principled methodologies for the systems engineering of artificial intelligence (AI).

The rest of this article is organized as follows. First, we provide background on transfer learning, domain adaptation, concept drift, PHM, and computer vision. We, then, justify the use of transfer distance as a metric by drawing from domain adaptation theory and present our methodology for quantifying transfer distance. Subsequently, we apply our methodology to characterize the transfer learning problems induced by an

Manuscript received 1 April 2021; revised 27 October 2021; accepted 12 December 2021. Date of publication 28 February 2022; date of current version 26 August 2022. This work was supported by the Naval Sea Systems Command under Grant N00024-17-C4008. *(Corresponding author: Tyler Cody.)*

The authors are with the National Security Institute, Virginia Tech, Arlington, VA 24061 USA (e-mail: tcody@vt.edu; sca2c@virginia.edu; pb3a@virginia.edu).

Digital Object Identifier 10.1109/JSYST.2022.3144837

actuator-rebuild procedure and mission deployment. Finally, Section VII concludes this article.

II. Background

We briefly review transfer learning, domain adaptation, concept drift, PHM, and computer vision, and note this article's relationship with them. In short, this article presents PHM and computer vision case studies in empirically characterizing transferability using principles from domain adaptation and methods from concept drift.

A. Transfer Learning

Transfer learning describes the idea of using knowledge from source systems to help learn in a particular target system. More formally, consider a learning problem that consists of a domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$ and a learning task $\mathcal{T} = \{\mathcal{Y}, P(Y|X)\}$, where $X = x$ for $x \in \mathcal{X}, Y = y$ for $y \in \mathcal{Y}$, and P denotes a probability distribution. Transfer learning uses knowledge from a source learning problem $\{\mathcal{D}_S, \mathcal{T}_S\}$ to improve the learning of a function $f_T : \mathcal{X}_T \to \mathcal{Y}_T$ in a target learning problem $\{\mathcal{D}_T, \mathcal{T}_T\}$, where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$ [2].

Transfer learning enables learning in environments where data are limited. Perhaps more importantly, it allows learning systems to propagate their knowledge forward through distributional changes, such as the degradation and wear of physical components, changes in use cases and functionality, and policy changes regarding the use of particular features $\mathcal{X}$ and labels $\mathcal{Y}$ [3]. The classical approaches to transfer learning involve selecting or weighing samples from the source, projecting the source and target features into a latent space, or bounding the parameters of the target model within a range of the source model's parameters [2].

Identifying whether or not transfer learning is an appropriate solution for a particular learning problem is crucial [4]. Failure to do so can result in *negative transfer*, wherein dissimilarity between the source and target systems results in transfer learning under-performing traditional machine learning approaches. While the extent of negative transfer is algorithm-dependent, the existence of negative transfer is tied to the distributions underlying the learning problem [5]. Thus, closeness between the source and target distributions is a precondition for transfer learning success.

B. Domain Adaptation

Domain adaptation is a subfield of transfer learning where $\mathcal{X}_S \times \mathcal{Y}_S = \mathcal{X}_T \times \mathcal{Y}_T$ [6]. In other words, only the probability distributions change between the sources and target, not their sample spaces. Domain adaptation theory places transfer distance at the center of bounding error in new environments [7], [8]. The common approach taken is to note that the error in the target is related to the error in the source plus some measure of similarity between the source and target.

These bounds can be loosely represented by the following inequality:

$$\epsilon_T \leq \epsilon_S + \delta + C \tag{1}$$

where ϵ_T and ϵ_S are the errors in the source and target, respectively, δ is the transfer distance between domain distributions, and C is a term that accounts for relevant complexities, e.g., VC-dimension [8] and sample size. Although inequality (1) is a rough approximation of the underlying learning theory, specifications can be added to arrive at proper, learning theoretic bounds using statistical divergence [7], $\mathcal{H}$-divergence [8], the Rademacher complexity [9], or integral probability metrics [10].

And, as Redko *et al.* [11] showed in their extensive survey of domain adaptation theory, ϵ_S and δ are the core terms in theoretical upper-bounds on error in new environments ϵ_T. Terms for capacity, sample size, information complexity, and others, provide nuance, but do not drive the upper-bound except in their extreme realizations. Moreover, inequality (1) emphasizes the term δ because it most directly couples systems-level design and operation to transferability and generalization difficulty of component-level learning systems. It is not clear that terms related to algorithm design and hypothesis class selection provide a similar mechanism.

C. Concept Drift

Whereas transfer learning considers distributional change between a source and target, concept drift considers distributional change that occurs in streaming data from one stable distribution, termed as a concept, to another. There are many metrics similar to transfer distance used in the concept drift literature to characterize drift [12]. Drift in these streaming systems has been modeled and simulated using the Gaussian mixture models (GMMs) [13], [14]. Many methods use the Hellinger distance to calculate distributional divergence because it is bounded $[0, 1]$ and symmetric [12], [15]. Consistent with concept drift literature, we use a combination of the GMMs and the Hellinger distance to characterize distributional change.

D. Prognostics and Health Management

PHM is concerned with the use of prognostics and diagnostics for the management of machine health [16]. In mechanized systems generally, it is essential for continuous operation, and, thus, is an important field of engineering research. As machine down-time is the eminent failure in production systems [17], PHM is crucial to economic productivity. Furthermore, PHM helps safeguard critical systems, such as gears in rotorcraft [18], whose failure can cause loss of propulsion mid-flight, and air filtration systems [19], whose failure in high pressure environments, such as submarines, can be equally catastrophic, among others [20].

Currently, machine health management is dominated by time-based maintenance schedules, however, there is an increasing interest in and use of data-driven PHM for adaptive scheduling [21]. This has led to extensive application of machine learning for health state classification and remaining useful life regression. There is a much smaller body of literature, however, using transfer learning to deal with the challenges these methods face in practice due to the aforementioned nonstationarities and label constraints [22]–[26]. While feature selection and metric learning use notions related to transfer distance and offer

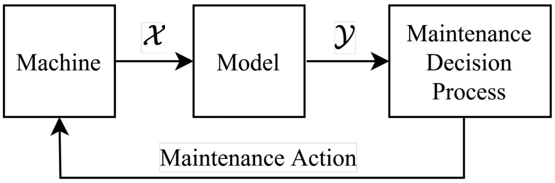


Fig. 1. Data-driven models inform maintenance actions, which change the distribution of their data. Nonstationarity is inherent in PHM.

promising directions for further development [27]–[29], they are still scoped to designing better learning algorithms, not to designing better systems for learning algorithms.

Nonstationarity is a fundamental challenge in PHM. In data-driven PHM, sensor data from machines are used for prognostics and diagnostics to inform operations management. When a maintenance action is taken, such as a machine rebuild, where the machine is deconstructed and rebuilt, the distribution of the sensor data changes. This cycle is shown in Fig. 1. Minor physical differences in the tensions of fasteners or locations of sensors can degrade predictive performance. The extent of degradation is difficult to ascertain because after an example of failure occurs, the system will be repaired, inducing a distribution change, or will be deemed irreparable.

Thus, in PHM systems, there is a real limit in our ability to address nonstationarity with algorithm design; it is necessary to take into account the role of system design in the generalization of learning. And to that end, it is necessary to have metrics, which can link notions like the design of maintenance procedures, e.g., regarding details like tensions and sensor locations, to notions like the transferability of knowledge.

In recent work, we extensively studied PHM for hydraulic actuators in order to better place related data-driven modeling in a systems context, including cost and power constraints [30]–[33]. These studies have used a fault-simulating test-bed that consists of two matched rotary actuators, where one acts as the actuator and the other acts as the load [34]. Here, we use data collected from this test-bed to extend the literature on data-driven PHM for hydraulic actuators by explicitly modeling the transfer distance associated with a rebuild procedure. Previously, we showed that sample transfer can be used to recover performance across the rebuild [35]. Here, instead of solving the transfer learning problem, in contrast, we use transfer distance as a means of characterizing the transfer learning problem associated with the rebuild.

E. Computer Vision

Computer vision is a broad field concerned with visual perception and pattern recognition. In recent years, deep learning has overtaken handcrafted feature engineering methods for processing images in the computer vision research literature [36], [37]. Instead of extracting expert-defined features from images as a preprocessing step, deep learning takes raw images as inputs and learns to both extract its own features and make predictions as part of a single, end-to-end process. While deep learning increases predictive performance and allows for novel use cases, it is heavily reliant on large datasets [38].

As previously described, in defense applications, this presents a bottleneck to deployment. Image classifiers have been trained to detect planes and their orientation when parked in airports' aprons using knowledge transferred from general visual recognition tasks [39]. However, such models are highly dependent on the airports included in training. As we will demonstrate, classifiers can suffer a decrease in performance when the biomes surrounding the airports change between training and operation. We calculate the transfer distance associated with transferring a model from one geographical region to another, as in a mission deployment scenario, and use it to anticipate model degradation. We use an autoencoder for dimension reduction, similar to existing approaches to an explainable AI [40].

An explainable AI seeks to alleviate challenges in AI assurance, including ethics, bias, fairness, and robustness, by making models, their training, and their inputs and outputs interpretable [41]. Deeper methodology concerns analysis of counterfactuals and causality, for example, in terms of graph neural networks [42]. Methods targeting robustness are varied and a subject of ongoing research [43], [44]. While robustness is typically framed in terms of stability against perturbation, the use of transfer distance herein alternatively concerns changes in stable points altogether, i.e., not in maintaining performance within a neighborhood of a particular system but rather between and across systems (due to life cycles, changes in use, etc.). Thus, transfer distance is relevant to traditional notions of robustness, but also applicable to assured and explainable AI broadly.

III. Methods

Transfer distance is usually referred to informally, e.g., to describe *near* or *far* transfer. It is implicit in the use of the Wasserstein distance [45], maximum mean discrepancy [46], [47], generative adversarial networks [48], [49], and others, to calculate distributional-divergence-based components of loss functions in transfer learning algorithms. We consider transfer distance explicitly, in a way that may not necessarily be useful in calculating loss functions, but is interpretable to system designers and operators. Transfer distance is defined as follows.

Definition 1 (Transfer Distance): The transfer distance between a source and target learning system, denoted by S and T, respectively, is a measure δ

$$\delta : P_S \times P_T \rightarrow \mathbb{R}$$

on probability measures P_S and P_T from the source domain $\mathcal{D}_S$ and task $\mathcal{T}_S$ and target domain $\mathcal{D}_T$ and task $\mathcal{T}_T$, respectively.

More general definitions are possible. This definition directs our interest toward the marginal distributions $P(X)$ from the domains $\mathcal{D}$ and posterior distributions $P(Y|X)$ from the tasks $\mathcal{T}$. For the purposes of explainability and analysis, we model these distributions explicitly, in closed-form, and take a Bayesian approach to constructing the posterior. We only fit $P(X|Y)$, and using an estimate for the prior $P(Y)$, construct the marginal $P(X)$ and posterior $P(Y|X)$.

Our algorithm for computing transfer distances can be described as follows. We assume that $\mathcal{X}_S = \mathcal{X}_T = \mathcal{X}$, $\mathcal{Y}_S = \mathcal{Y}_T = \mathcal{Y}$, that $\mathcal{X}$ is continuous, and that $\mathcal{Y}$ is discrete. We first fit the likelihood distributions $P_S(X|Y = y)$ and $P_T(X|Y = y)$ for

Algorithm 1: Calculating Transfer Distance in Domain Adaptation with Discrete $\mathcal{Y}$.

```
Input: $data_S$, $data_T$, $P(Y = y) \forall y \in \mathcal{Y}$
Output: $\delta_{X|Y=y}, \delta_X, \delta_{Y=y|X}$
def fit(data):
    $\{P(X|Y = y)\}_{y \in \mathcal{Y}} \leftarrow$ fitter$(data)$
    $P(X) \leftarrow \sum_{y \in \mathcal{Y}} P(X|Y = y)P(Y = y)$
    $\{P(Y = y|X)\}_{y \in \mathcal{Y}} \leftarrow \{P(X|Y = y)P(Y = y)/P(X)\}_{y \in \mathcal{Y}}$
    return $P(X|Y), P(X), P(Y|X)$

$\{P_S(X|Y = y)\}_{y \in \mathcal{Y}}, P_S(X), \{P_S(Y = y|X)\}_{y \in \mathcal{Y}} \leftarrow$ fit$(data_S)$
$\{P_T(X|Y = y)\}_{y \in \mathcal{Y}}, P_T(X), \{P_T(Y = y|X)\}_{y \in \mathcal{Y}} \leftarrow$ fit$(data_T)$

$\{\delta_{X|Y=y}\}_{y \in \mathcal{Y}} \leftarrow \delta(P_S(X|Y), P_T(X|Y))$
$\delta_X \leftarrow \delta(P_S(X), P_T(X))$
$\{\delta_{Y=y|X}\}_{y \in \mathcal{Y}} \leftarrow \delta(P_S(Y|X), P_T(Y|X))$

return $\{\delta_{X|Y=y}\}_{y \in \mathcal{Y}}, \delta_X, \{\delta_{Y=y|X}\}_{y \in \mathcal{Y}}$
```

all $y \in \mathcal{Y}$. We construct $P_S(X)$ and $P_T(X)$ using a prior $P(Y)$ and the total probability law, and then construct $P_S(Y = y|X)$ and $P_T(Y = y|X)$ for all $y \in \mathcal{Y}$ using the Bayes theorem. We, then, sample from $\mathcal{X} \times \mathcal{Y}$ according to the source and target distributions and calculate the transfer distance δ using these samples. This process is shown in Algorithm 1.

We use the GMMs to fit the likelihoods $P(X|Y)$, i.e., as the *fitter* method in *fit* function of Algorithm 1. The Gaussian mixture modeling is a clustering technique whereby a mixture of probability weighted multivariate Gaussian distributions is fit to data. Each point is assigned to a single multivariate Gaussian, i.e., its cluster. For a GMM with K clusters

$$p(X) = \sum_{k=1}^{K} \pi_k \mathcal{N}(X|\mu_k, \sigma_k)$$

where $p(X)$ is the density function of X, π_k is the probability weight of cluster k, and $\mathcal{N}$ is the multivariate Gaussian distribution with mean μ_k and covariance σ_k.

Here, an empirical prior probability—the ratio of the sample size of each label to the total number of samples—is used to estimate $P(Y = y)$ and construct $P(X)$ from the GMMs of $P(X|Y)$, except for the binary classification problem in PHM where sensitivity to $P(Y = y)$ is considered. Methods for estimating $P(Y = y)$ are historic, varied, and often application-specific [50]. Our use of empirical priors is not a recommendation, per se, and choice of an alternative approach is a means by which practitioners can tailor their use of the transfer distance methodology to their specific application.

Explicitly, closed-form models of the source and target allow for a rich set of distance functions. Different applications may call for different distances, and closed-form distributions afford this flexibility. In our case, we use the Hellinger distance and the Kullback–Leibler (KL) divergence as our transfer distances δ. For the sake of computation, distributions are discretized by sampling a region defined over the union of the supports of the distributions in question.[1] Given two discrete probability distributions $P = (p_1, \ldots, p_n)$ and $Q = (q_1, \ldots, q_n)$, the Hellinger distance between P and Q is

$$H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{n} (\sqrt{p_i} - \sqrt{q_i})^2}.$$

H is symmetric and bounded $[0, 1]$, where $H = 0$ implies that the distributions are completely identical and $H = 1$ implies that they do not overlap at all. The KL divergence between P and Q is

$$KL(P, Q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i}.$$

The KL is not symmetric and is unbounded above $[0, \infty)$, where its lower bound implies that the distributions are completely identical. The Hellinger distance is used for the PHM case studies in system design and the KL divergence is used for the computer vision case studies in system operation. The reason for changing distance measures is to underscore the generality of the proposed methodology.

IV. Transfer Distance for System Design

In system design, transfer distance can be used to design systems with an awareness of the generalization difficulty faced by component learning systems. Generalization difficulty concerns the difficulty of achieving a certain level of error on new data. Different design decisions can be associated with different generalization difficulties. inequality (1), i.e.,

$$\epsilon_T \leq \epsilon_S + \delta + C$$

suggests that a higher transfer distance δ is associated with a higher demand on the error in the source ϵ_S and the term C to keep the upper bound on error in the target ϵ_T the same as

[1] Near-zero values are treated as zero to bound the supports.

with a lower transfer distance. Therefore, transfer distance has a strong, fundamental influence on generalization difficulty.

In cases where the distance between the source and target is, for example, associated with some physical change in the system, we can use transfer distance as a means of associating the physical change with generalization difficulty. Consider the generalization of prognostics models across system rebuilds. In previous work, we found that while binary health states for hydraulic actuators can be classified with an accuracy of 98% when trained and tested on the same actuator, but when the actuator is deconstructed and rebuilt, the same classifier does marginally better than random guessing [35]. When transfer learning is applied, classification accuracy recovers to almost 90%.

In the following, transfer distance is used to characterize the generalization difficulty associated with a particular actuator-rebuild procedure. We show, how an analysis of transfer distance can be used to understand why the original classifier failed, to suggest why transfer learning worked, and ultimately, to inform the iterative design of rebuild procedures to limit degradation in predictive performance across system rebuilds. We quantify the generalization difficulty associated with the rebuild procedure in terms of the transfer distance between binary and multiclass health state classification before and after the rebuild. Then, we quantify the number of samples required to achieve a stable estimate of transfer distance.

Faults were simulated on a hydraulic actuator, the actuator was deconstructed and rebuilt, and the faults were resimulated. The failure modes considered are opposing load, external load, bypass valve, and leak valve failures, among miscellaneous others. The hydraulic-actuator test stand is equipped with sensors to collect acceleration, pressure, flow, temperature, and rotary position. In preprocessing, to capture aspects of time-dependence, the data are first windowed and summarized by the mean and standard deviation of each window. Then, to reduce the dimension of the data, principal component analysis is applied. The first two principal components capture 90% of the variance in the windowed features. These two components are used in our studies.

A. Transfer Distance Induced by Rebuild

First, we consider binary health state classification, where we learn to predict whether the hydraulic actuator is healthy, $Y = 0$, or damaged, $Y = 1$. The original actuator is the source, the rebuilt actuator is the target, and we are interested in empirically quantifying the change in the binary classification problem induced by the rebuild process, i.e., the changes in the distributions underlying the problem. There are 789 healthy and 1480 damaged samples in the source, and 1098 healthy and 1822 damaged samples in the target.

The empirical prior $P(Y = 0)$ is given by the ratio of healthy samples to damaged samples, but such a prior implies almost even–odds of failure. We approximate the empirical prior as $P(Y = 0) = 0.40$ and compare against $P(Y = 0) \in \{0.9, 0.99, 0.999\}$. The same priors are used for both the source and target.

TABLE I
HELLINGER TRANSFER DISTANCE FOR RELEVANT DISTRIBUTIONS

Transfer Distance	$P(Y = 0)$			
	0.40	0.90	0.99	0.999
$\delta_{X\|Y=0}$	0.22	-	-	-
$\delta_{X\|Y=1}$	0.54	-	-	-
δ_X	0.41	0.25	0.22	0.22
$\delta_{Y=0\|X}$	0.24	0.23	0.23	0.23

Note: $\delta_{X|Y}$ does not depend on $P(Y)$.

The fitted likelihoods and the constructed posterior probability of being healthy are plotted in Fig. 2. The likelihood densities in Fig. 2(a) and (b) show the source in red and target in blue, fit with 2-component GMMs, where each concentric ellipse represents one standard deviation from a component's mean. The plotted points are from samples held-out from the fitting process. Whereas the healthy densities overlap closely between the source and target, the damaged densities do not. The target, rebuilt actuator has a larger spread in the distribution of damaged data when represented by its first two principal components. Classification likely dropped because of this increased variance. Despite this difference, the posteriors, shown in Fig. 2(c) for $P(Y = 0) = 0.40$, are fairly similar. Transfer learning likely succeeded at bringing accuracy back to nearly 90% because the increased variance in the damaged likelihood did not strongly affect the posterior.

Transfer distances δ are shown in Table I. As in the plots, the healthy likelihoods are closer than the damaged likelihoods. Notably, the transfer distance between the marginals $P(X)$ is larger than that between the posteriors $P(Y = 0|X)$. In other words, there are changes in the distribution of the sensor data that do not have a material effect on the binary classification problem. We can also note that as the prior odds of failure decrease, δ_X and $\delta_{Y=0|X}$ decrease as well, because the difference in the damaged likelihood is weighted less.

These results show that the rebuild procedure affects the distributions of damaged data far more than the distribution of healthy data. This means that while healthy behavior appears similar across rebuilds, failure does not. This is particularly worrisome because in fielded systems we will typically only have access to healthy samples. The transfer distance between the healthy source and target data suggests a much smaller change than that actually occurs. This finding reaffirms our position that designing systems to avoid difficult transfer learning problems is essential to AI engineering because there are distributional changes over a system's life cycle that we cannot sample and empirically characterize in the field.

In PHM systems, it may be the case that some failure modes are similar across many machines or many rebuilds, whereas others are not. Transfer distance provides a means for empirically quantifying how transferable failure modes are relative to each other, and thereby serves as a mechanism for directing related engineering effort, such as data collection and algorithm design.

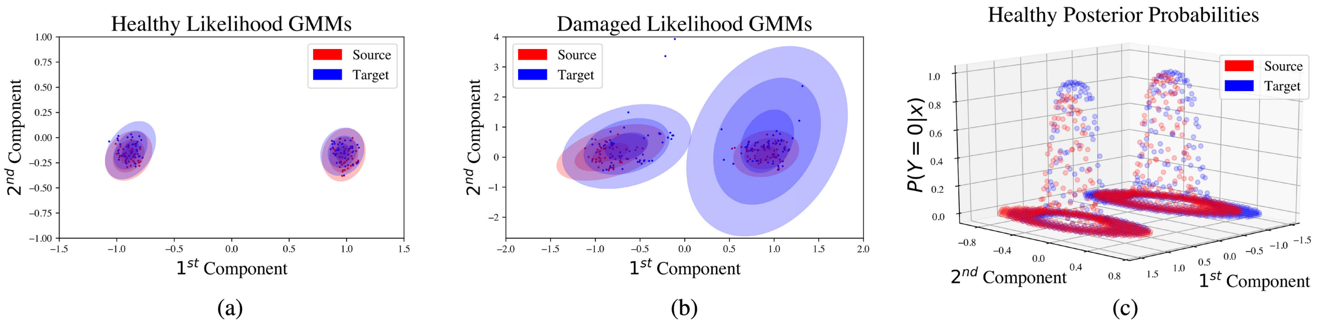


Fig. 2. Likelihood densities and healthy posterior distributions. Each point corresponds to a single instance of windowed sensor-features. (a) $p(X|Y = 0)$. (b) $p(X|Y = 1)$. (c) $P(Y = 0|X)$.

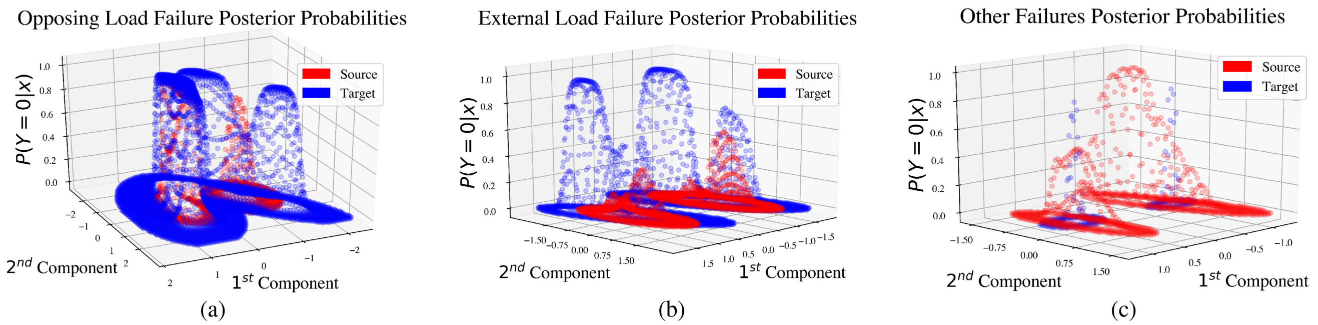


Fig. 3. Posterior distributions for different failure modes. Each point corresponds to a single instance of windowed sensor-features. (a) $p(X|Y = 1)$. (b) $p(X|Y = 2)$. (c) $P(Y = 5|X)$.

TABLE II
HELLINGER TRANSFER DISTANCE FOR RELEVANT DISTRIBUTIONS

Failure Type	Likelihood $\delta_{X\|Y}$	Posterior $\delta_{Y\|X}$
Opposing Load	0.53	0.64
External Load	0.41	0.72
Bypass Valve	0.18	0.69
Leak Valve	0.74	0.88
Other	0.67	0.80

Since transfer learning comes with associated costs and risks, it is important to know where it is needed and where it is not. A need-based approach not only allows for reduced knowledge transfer and retraining, but it also allows transfer learning algorithms to specifically focus on transferring knowledge for those failure modes that need source knowledge the most.

We quantify the transfer distance between failure modes in the source and target using a multiclass health state classification problem. Now, $\mathcal{Y} = \{0, 1, 2, 3, 4, 5\}$ where $Y = 0$ signifies healthy and $Y = 1, \ldots, 5$ signify opposing load, external load, bypass valve, leak valve, and other failures, respectively. We have a similar number of samples between source and target and across failure modes. Using the presented methodology we fit a posterior distribution $\forall y \in \mathcal{Y}$. Table II shows the likelihood and posterior transfer distances for each failure mode.

Opposing load failures have a posterior transfer distance of 0.64 and leak valve failures have a posterior transfer distance of 0.88. This suggests that the sensor-data representations of opposing load failures in the source and target actuators are closer than those of leak-valve failure. Put flatly, opposing-load failures look more similar after the rebuild than leak-valve failures.

Fig. 3 shows the source and target posterior probabilities for opposing load, external load, and other miscellaneous failures. The overlap of the distributions in the plots corresponds to the posterior transfer distances in Table II. Perhaps an algorithm designer may conclude that knowledge transfer is feasible for opposing load failures, but not for other failures. Or, perhaps a systems engineer would suggest redesigning the rebuild procedure to bring those failure modes with a higher transfer distance closer in the PCA space.

B. Transfer Distance and Sample Size

We have shown how transfer distance can be used to characterize transferability and provide insights for system and algorithm design. It is important to note that the distribution of the target actuator has a certain sample complexity. Transfer learning that relies on measures of distributional difference should wait for the distribution to settle first; otherwise, methods, such as sample weighting and selection, will be using inaccurate estimates of distributional divergence. Similarly, transfer distance may require a number of samples to be collected before it can be considered a reliable metric for design and operational decision-making.

In the hydraulic actuators, each sensor-feature, e.g., the mean of acceleration 1, the standard deviation of pressure 1, etc., has its

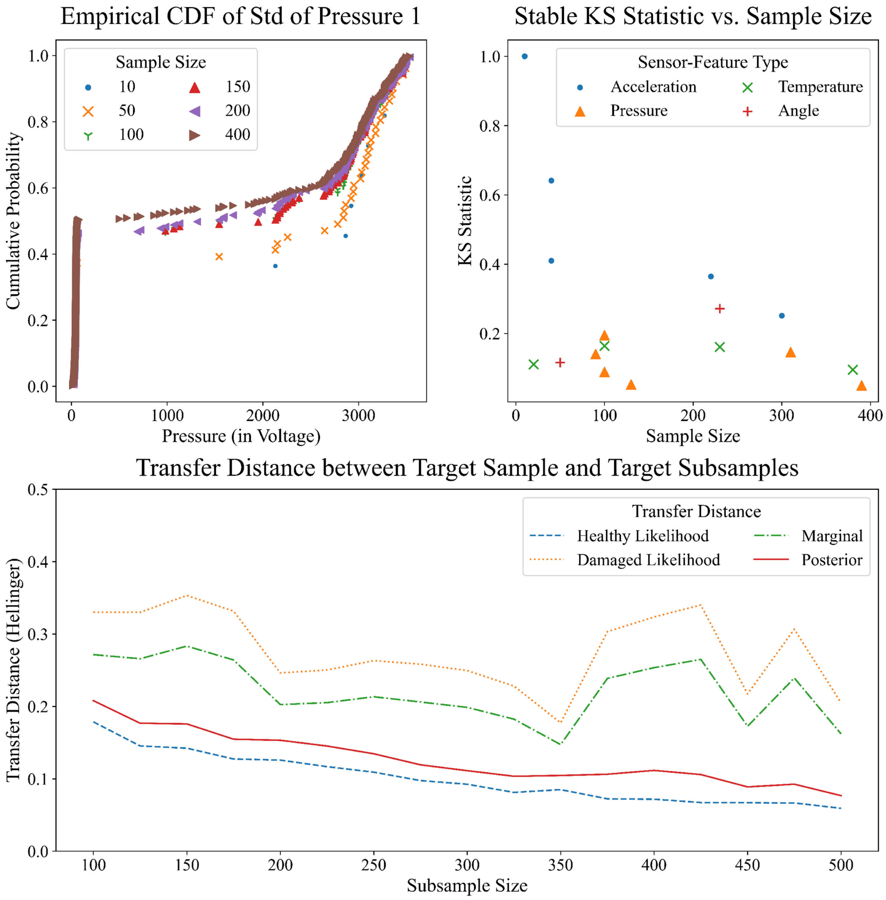


Fig. 4. Top-left plot shows how the empirical CDF of the standard deviation of a pressure gauge changes with sample size. The top-right plot shows how many samples it takes for different sensor-feature types to converge to a stable value, labeled according to sensor-type. The bottom plot shows the transfer distance between GMMs trained on a subsample of target data and a full sample of target data.

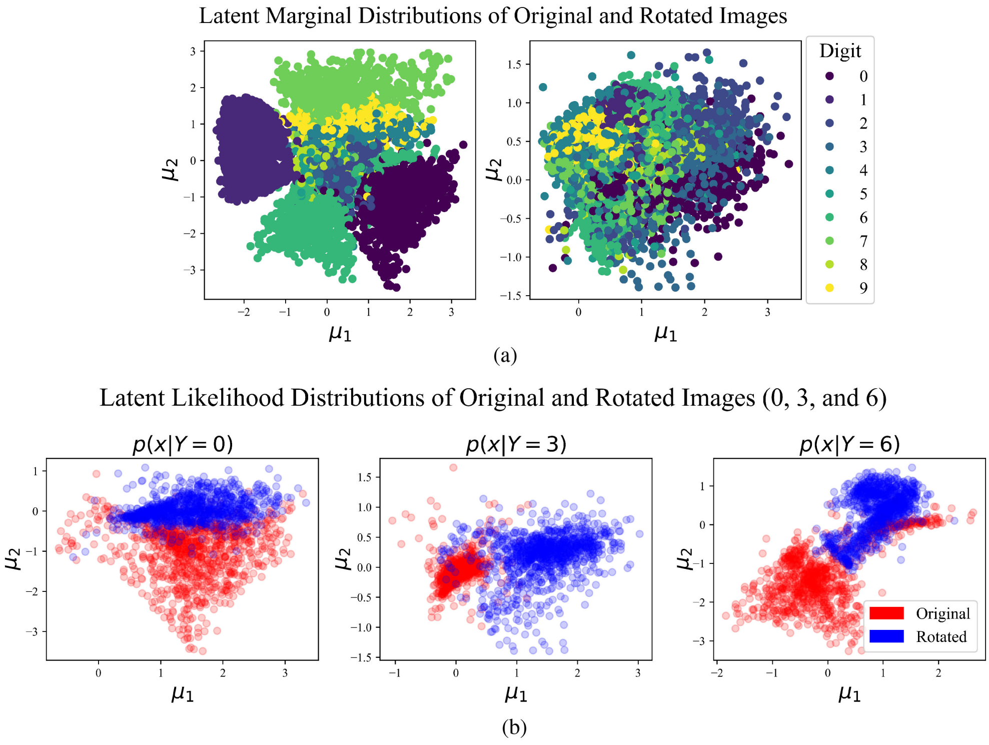
own sample complexity. Note the top-left plot in Fig. 4, which shows the empirical cumulative distribution functions (CDFs) associated with different size samples of the standard deviation of a pressure gauge. The CDF appears not to settle until 150 to 200 samples. If we use the Kolmogorov–Smirnov (KS) statistic, which gives the largest absolute difference between two univariate CDFs, we can test when successive increases in sample size no longer change the distance between a sensor-feature's CDF in the source and target. In the top-right plot of Fig. 4, the point where the change in the KS statistic between the source and target for successive sample sizes changes less than 5% is plotted for each type of sensor-feature, e.g., acceleration, pressure, etc. Apparently, the distances between the source and target univariate CDFs converge at different rates. Accelerations have the largest KS statistics, but also the lowest sample size to settle.

We are learning using multiple sensor-features, thus, we are interested in how they settle jointly. In the bottom plot of Fig. 4, we consider sensor-feature interdependence by calculating the Hellinger transfer distances δ between target subsamples of a size corresponding to the x-axis and the full target sample. Transfer distances $\delta_{Y|X}$ and $\delta_{X|Y=0}$ decrease as sample size increases, and transfer distances δ_X and $\delta_{X|Y=1}$ roughly follow the same trend. Based on these results, it appears as though it takes at least 300 to 350 samples in the target before estimates of distributional divergence are stable. Note, that in practice, we often will only be able to conduct this analysis using the healthy data.

In the context of machinery, depending on the nature of a maintenance procedure, the time to estimate the new distribution of sensor-data may change. This period relates to the lag-time before we can transfer knowledge to the new system to support data-driven PHM. The design of maintenance procedures to influence the length of this intervention is an important aspect of keeping PHM systems functioning.

V. Transfer Distance for System Operation

In system operation, transfer distance can be used to operate systems with an awareness of the expected generalization performance of component learning systems. Generalization performance concerns a learning system's error on new data. Different operational decisions are associated with different expected generalization performances. Inequality (1) suggests that transfer distance plays a fundamental role in determining the upper bound on error in new environments. Therefore, transfer distance has a strong connection to expected generalization performance.



(a)

(b)

Fig. 5. MNIST in latent space. (a) MNIST original, source images (left) and rotated, target images (right) in the variational autoencoder's latent space. (b) The variational autoencoder's latent space shows the effect of rotation varies by digit.

In defense applications of computer vision, look angle, pixel density, time of day, and biome, for example, can vary between missions. Even when the sample spaces of images $\mathcal{X}$ and image labels $\mathcal{Y}$ have the same structure, the probability distributions associated with those sample spaces can differ drastically. Sometimes, one can intuit the existence of significant differences, for example, between image classification problems in the tundra and jungle. Other times, it is not as clear, for example, between classification problems in the Southern and Northern California. In either cases, transfer distance can empirically support or reject such intuition.

In the following, we first explore the relationships between transfer distance and expected generalization performance on the canonical handwritten digit recognition dataset MNIST [51]. Then, with this understanding, we explore an application in defense where a model trained to detect the presence of aircraft in the Southern California is deployed on a mission in the Northern California [52]. In both cases, we use autoencoders to compress the images into a low-dimensional, latent representation before applying the Algorithm 1 to compute transfer distances of interest. We use the GMMs as before, but now use the KL divergence instead of the Hellinger distance as our measure of transfer distance δ.

A. *MNIST and Expected Operational Performance*

Just as transfer distance can be used as a metric for assessing the difficulty of generalization associated with a particular system design, it can be used to assess expected operational performance. Unlike in system design, in system operation we do not have direct control over transfer distance. We are not looking to change transfer distance directly, but rather, to operate in such a way that performance remains satisfactory.[2] Viewed discretely, we have a training environment, the source, and an operating environment, the target, and are interested in identifying if generalization performance in the operating environment will be satisfactory.

To see how transfer distance relates to expected operational performance, consider the MNIST handwritten digit recognition problem. The dataset contains examples of handwritten digits 0 through 9. We let the original data act as the source, training environment. To create a target we rotate all original data by 90-degrees clockwise. We fit a variational autoencoder to the source images and use it to represent the source and target images as bivariate Gaussian distributions [53].

The transformed images are plotted in Fig. 5(a) according to their Gaussian means μ. Whereas 0, 1, and 6 are well separated in the source, as shown in the left plot, no rotated digits are well separated in the target, as shown in the right plot. The target images are interspersed with each other and have a smaller variance in μ_1 and μ_2 than the source images. This immediately suggests that the rotation of the images has a significant effect on $P(X)$.

This difference in $P(X)$ is not the same for all digits, however. Consider the digits 0, 3, and 6, as shown in Fig. 5(b). While all show differences, both the source and target "0" and "6" images share some overlap. In contrast, the source and target "3" images

[2] In general, design and operation are inextricable, but herein we establish a dichotomy to emphasize the dual use of transfer distance.

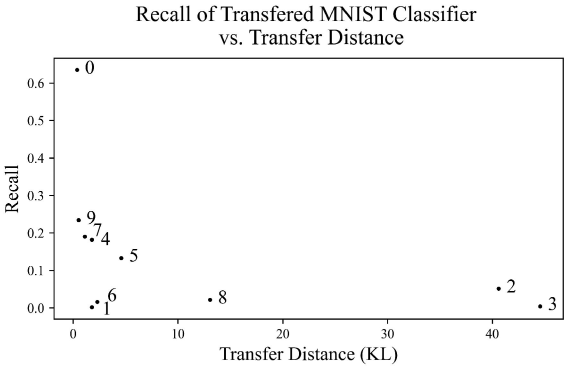


Fig. 6. Higher transfer distance digits have low recall.

are almost partitioned by $\mu_1 = 0.5$. It makes intuitive sense that 0 and 6 are more similar because of the invariance of circles to rotation.

To investigate further, we use a random forest to classify digits [54]. When we calculate the recall on the rotated, target images of a classifier trained on the nonrotated, source images we find that those digits with a higher transfer distance (in this case a higher KL divergence) have a lower recall, as shown in Fig. 6. Different to accuracy, recall considers the true positive rate, i.e., the ratio of correct classifications to number of instances of that class. "0" images have the highest recall and lowest transfer distance. And while digits "1" and "6" have low recall and low transfer distance, there are no cases of high transfer distance and high recall. Thus, it seems high transfer distances have low recall. Given a measurement of transfer distance, we can form an empirical judgement of expected operational performance and, correspondingly, can make empirically informed operational decisions. In the following, we consider a "go, no go" mission deployment problem in aircraft detection.

B. Mission Scenario in Aircraft Detection

Object detection from overhead imagery is a core function in defense systems. Despite the success of high-capacity models, like deep learning, image classifiers are not global. Classifiers trained in one geographic region suffer performance degradation when deployed in other geographic regions. Fundamentally, this occurs because of a change in the underlying distribution of images. Transfer distance can be used to anticipate and detect drops in performance by comparing the distributional difference between samples from the training and operating environments.

Consider a case where a classifier is trained to detect the presence of aircraft in the Southern California and is tasked with operating in the Northern California. Example images are shown in Fig. 7. There are roughly 20 000 images from the Southern California and 12 000 images from the Northern California. We trained a convolutional neural network to detect aircraft on Southern California images.

When classifying held-out images from the Southern California the classifier's accuracy is nearly 98%, but when classifying images from the Northern California accuracy drops to nearly 85%, as shown in Fig. 8. The classifier still has predictive power, but, in critical applications like defense, the difference between a 2% error rate and a 15% error rate is significant enough to constitute failure.

Fig. 7. Example aircraft images and nonaircraft images from California in the top and bottom rows, respectively.

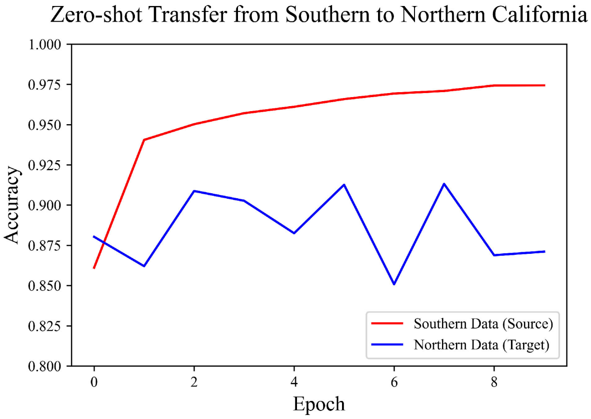
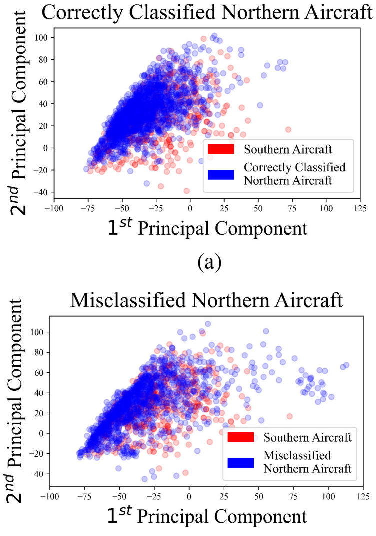


Fig. 8. Shown is the classification accuracy of a convolutional neural network trained in the Southern California evaluated on the Southern California images, in blue, and on the Northern California images, in orange, over the course of ten training epochs.

In order to apply our transfer distance methodology, we first train an autoencoder on the Southern California images. To do this, we initialize a convolutional autoencoder with weights from the VGG-16 image classification network and then we fine-tune those weights [55]. We use the autoencoder to encode the images into vectors. Then, we find the principal components of the encoded Southern California images and transform all images into the first two principal components, as in the actuator example. In contrast to the actuator example, however, because of the size of the dataset, we batch the data into samples of 100 before fitting GMMs.

When we calculate transfer distances δ_X between samples drawn from the Southern California, we find them to have a mean KL divergence of 5.60. When we calculate transfer distances δ_X between samples drawn from the Southern and Northern California, we find them to have a slightly higher mean KL divergence of 5.97. This suggests that samples drawn from the Southern and Northern California are, on average, farther from each other than two samples drawn from the Southern California. The small difference in expected transfer distance corresponds to the slight drop in the classification accuracy in Fig. 8. While this transfer distance may seem small, it highlights a general difference between the Southern and Northern California images for deeper analysis. We can investigate further by calculating transfer distances $\delta_{X|Y}$ of correctly and incorrectly classified images.



(a)

(b)

Fig. 9. First two principal components of true positives and false negative classifications when classifying images in the Northern California using a classifier trained in the Southern California. Misclassified Northern California aircraft images do not share a center of mass with the Southern California aircraft images. (a) True Positives. (b) False Negatives.

Correctly classified Northern California aircraft images have a KL divergence of 0.94 from Southern California aircraft images, while misclassified Northern aircraft images have a KL divergence of 1.99, twice as high. These distances correspond to true positive and false negative cases, respectively. Correctly classified nonaircraft images from the Northern California have a KL divergence of 2.34 from Southern California nonaircraft images, while misclassified nonaircraft images from the Northern California have transfer distance of 3.11. Note, the transfer distance for incorrectly classified images is higher than the transfer distance for correctly classified images for both aircraft and nonaircraft images. That is, higher transfer distance correlates to higher error. We can analyze why this is so by using the principal components of the encoded images.

True positives refer to correctly classified aircraft images and false negatives refer to incorrectly classified aircraft images. The true positives and false negatives associated with the classifier trained on the Southern California overhead imagery are shown in Fig. 9(a) and (b), respectively. Notice that the correctly classified aircraft images are near the center of mass of the Southern California aircraft images while the incorrectly classified aircraft images are not. In other words, the incorrectly classified Northern California aircraft images are in the tails of the distribution of the Southern California aircraft images.

This suggests that system operators can empirically inform "go, no go" deployment decisions using transfer distance. In this case, the transfer distance between unlabeled images δ_X suggests a slight drop in the performance. Further, transfer distance between misclassified images is higher than that of the correctly classified images. Before deployment, system operators can use this empirical evidence to anticipate challenges to mission success. After deployment, system operators can use transfer distance to adjust their confidence in the model's classification accuracy in real-time.

VI. Remarks and Limitations

In the preceding, we demonstrated how to use transfer distance to compare the transferability of binary and multiclass health state classifiers. In doing so, we showed how transfer distance can be used to quantify the transferability of both generalized and specific modes of failure across maintenance procedures. In addition, we showed how to determine the number of samples needed for stable estimates of transfer-distance and transfer-learning parameters, and discussed the role of the design of maintenance procedures in the length of this intervening period. We also demonstrated transfer distance's use in computer vision. In particular, we identified what kind of images are least transferable across changes in look angle and we anticipated and analyzed degradation in aircraft detection performance between geographic regions. We used different measures of transfer distance and generalization performance as well as different sized data-sets from different domains, i.e., sensor data and images, to highlight the generality of the presented transfer distance methodology. The varied concerns of these case studies may appear disparate when viewed bottom-up, but, when viewed top-down, transfer distance presents itself as an application-agnostic methodology for systems engineering.

The presented systems engineering methods are complementary to algorithmic methods. They are necessary in the many cases where data will not be available to sufficiently adapt component learning systems. They are complementary because robust and sample-efficient learning algorithms require less consideration in system design and operation, and more consideration in system design and operation requires less robustness and sample-efficiency from learning algorithms.

While the calculation of transfer distance using the Bayesian method presented herein is not computationally efficient, the purpose of transfer distance as a formal concept is to elevate its use beyond the definition of loss functions for machine learning. The Bayesian formulation emphasizes the varied roles of all the terms in the Bayes theorem, trading off computational efficiency for richness of information. Negotiating this tradeoff as appropriate for a given application and its available data are left to practitioners.

It is important to note that the full Bayesian characterization of transfer distance is not needed to use transfer distance to inform decisions in system design and operation. As mentioned, domain adaptation theory for upper-bounding error in new environments is built around the use of marginal distributions $P(X)$, i.e., unlabeled data. In addition, the distributions of the domain and task do not necessarily need to be fit empirically. Expert knowledge, e.g., in the form of physical models or mission profiles, can provide physics-based or judgmental models of distributions. And so, the data used to calculate transfer distance do not need to be the same as the data used by transfer learning algorithms, therefore, transfer distance can be estimated during operation in cases with insufficient data for transfer learning.

VII. Conclusion

As machine learning is deployed into systems, it is important to consider the role systems engineering plays as a mechanism for generalization. Systems engineering for AI requires metrics that can relate learning-theoretic concerns to the systems-level. Transfer distance is such a metric. In learning theory, it is central to the bounding of prediction error of learned models in new settings, such as rebuilt actuators or new look angles. At the systems-level, it serves as a measurement of the closeness of learning problems, and thereby a metric for designing and operating systems with the generalization performance of component learning systems in mind.

Herein, we formally defined transfer distance as a measure, presented an algorithm for calculating it, and demonstrated its use in system design and operation. We emphasized how, by using transfer distance as a metric, systems can be designed to influence generalization difficulty and can be operated to influence generalization performance. Better matching system design and operation with component learning systems means a lower chance of negative transfer, or at least a heightened ability to anticipate negative transfer, less frequent occurrences of drift or more anticipatable drift, and, overall, a lower burden on algorithmic robustness.

In future work, we plan to further explore the use of transfer distance in engineering practice. For example, in designing rebuild procedures, we aim to characterize the sensitivity of transfer distance to the tensions of fasteners, locations of sensors, and the manufacturer of replacement parts. Also, in making "go, no-go" operational decisions, e.g., in unmanned aerial systems, we aim to tie mission success to the transfer distance between training and operating environments. Lastly, the emphasis on problem domain and task in the definition of transfer distance follows from the status quo focus on problem solving in the machine learning literature. Future work should investigate the empirical use of transfer distancex between learning algorithms and their systems more broadly.

Acknowledgment

The authors would like to thank Luna Innovations, Inc. for providing a portion of the data used within this article. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Naval Sea Systems Command. Also, an early draft of the presented results is available on the *arXiv* [56].

References

[1] C. Rogers *et al.*, "Adversarial artificial intelligence for overhead imagery classification models," in *Proc. IEEE Syst. Inf. Eng. Des. Symp.*, 2019, pp. 1–6.

[2] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2009.

[3] T. Cody, S. Adams, and P. A. Beling, "A systems theoretic perspective on transfer learning," in *Proc. IEEE Int. Syst. Conf.*, 2019, pp. 1–7.

[4] M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich, "To transfer or not to transfer," in *Proc. Int. Conf. Neural Inf. Process. Syst. Workshop Transfer Learn.*, 2005, pp. 1–4.

[5] Z. Wang, Z. Dai, B. Póczos, and J. Carbonell, "Characterizing and avoiding negative transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11293–11302.

[6] J. Jiang, "A literature survey on domain adaptation of statistical classifiers," 2008. [Online]. Available: http://sifaka. cs. uiuc. edu/jiang4/ domainadaptation/survey

[7] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 129–136.

[8] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1, pp. 151–175, 2010.

[9] M. Mohri and A. Rostamizadeh, "Rademacher complexity bounds for non-I.I.D processes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1097–1104.

[10] C. Zhang, L. Zhang, and J. Ye, "Generalization bounds for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 3320–3328.

[11] I. Redko, E. Morvant, A. Habrard, M. Sebban, and Y. Bennani, "A survey on domain adaptation theory," 2020, *arXiv:2004.11829*.

[12] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean, "Characterizing concept drift," *Data Mining Knowl. Discov.*, vol. 30, no. 4, pp. 964–994, 2016.

[13] J. Wu, X.-S. Hua, and B. Zhang, "Tracking concept drifting with Gaussian mixture model," *Int. Soc. Opt. Photon. Vis. Commun. Image Process.*, vol. 5960, 2005, Art. no. 59604L.

[14] J. Diaz-Rozo, C. Bielza, and P. Larrañaga, "Clustering of data streams with dynamic Gaussian mixture models: An IoT application in industrial processes," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3533–3547, Oct. 2018.

[15] G. Ditzler and R. Polikar, "Hellinger distance based drift detection for nonstationary environments," in *Proc. IEEE Symp. Comput. Intell. Dyn. Uncertain Environ.*, 2011, pp. 41–48.

[16] K. L. Tsui, N. Chen, Q. Zhou, Y. Hai, and W. Wang, "Prognostics and health management: A review on data driven approaches," *Math. Problems Eng.*, vol. 2015, pp. 1–17, 2015.

[17] J. Li and S. M. Meerkov, *Production Systems Engineering*. Berlin, Germany: Springer, 2008.

[18] I. R. Delgado, P. J. Dempsey, and D. L. Simon, *A survey of current rotorcraft propulsion health monitoring technologies*. Cleveland, OH, USA: Nat. Aeronaut. Space Admin., Glenn Res. Center, 2012.

[19] F. Landolsi, H. Jammoussi, and I. Makki, "Air filter diagnostics & prognostics in naturally aspired engines," in *Proc. IEEE Int. Conf. Prognostics Health Manage.*, 2017, pp. 61–65.

[20] M. Eftekhari, M. Moallem, S. Sadri, and M.-F. Hsieh, "Online detection of induction motor's stator winding short-circuit faults," *IEEE Syst. J.*, vol. 8, no. 4, pp. 1272–1282, Dec. 2013.

[21] W. Zhang, D. Yang, and H. Wang, "Data-driven methods for predictive maintenance of industrial equipment: A survey," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2213–2227, Sep. 2019.

[22] F. Shen, C. Chen, R. Yan, and R. X. Gao, "Bearing fault diagnosis based on SVD feature extraction and transfer learning classification," in *Proc. IEEE Prognostics Syst. Health Manage. Conf.*, 2015, pp. 1–6.

[23] J. Xie, L. Zhang, L. Duan, and J. Wang, "On cross-domain feature fusion in gearbox fault diagnosis under various operating conditions based on transfer component analysis," in *Proc. IEE Int. Conf. Prognostics Health Manage.*, 2016, pp. 1–6.

[24] W. Lu, B. Liang, Y. Cheng, D. Meng, J. Yang, and T. Zhang, "Deep model based domain adaptation for fault diagnosis," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2296–2305, Mar. 2017.

[25] W. Zhang, G. Peng, C. Li, Y. Chen, and Z. Zhang, "A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals," *Sensors*, vol. 17, no. 2, pp. 425–445, 2017.

[26] X. Li, Y. Hu, M. Li, and J. Zheng, "Fault diagnostics between different type of components: A transfer learning approach," *Appl. Soft Comput.*, vol. 86, 2020, Art. no. 105950.

[27] S. Uguroglu and J. Carbonell, "Feature selection for transfer learning," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2011, pp. 430–442.

[28] X. Zhong, S. Guo, H. Shan, L. Gao, D. Xue, and N. Zhao, "Feature-based transfer learning based on distribution similarity," *IEEE Access*, vol. 6, pp. 35551–35557, 2018.

[29] Y. Xu *et al.*, "A unified framework for metric transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1158–1171, Jun. 2017.

[30] S. Adams *et al.*, "A comparison of feature selection and feature extraction techniques for condition monitoring of a hydraulic actuator," in *Proc. Annu. Conf. Prognostics Health Manage. Soc.*, 2017.

[31] R. Meekins *et al.*, "Cost-sensitive classifier selection when there is additional cost information," in *Proc. Int. Workshop Cost-Sensitive Learn.*, 2018, pp. 17–30.

[32] K. M. Farinholt *et al.*, "Developing health management strategies using power constrained hardware," in *Proc. Annu. Conf. Prognostics Health Manage. Soc.*, 2018, vol. 10, no. 1.

[33] S. Adams *et al.*, "Hierarchical fault classification for resource constrained systems," *Mech. Syst. Signal Process.*, vol. 134, 2019, Art. no. 106266.

[34] S. Adams, P. A. Beling, K. Farinholt, N. Brown, S. Polter, and Q. Dong, "Condition based monitoring for a hydraulic actuator," in *Proc. Annu. Conf. Prognostics Health Manage. Soc.*, 2016.

[35] T. Cody *et al.*, "Transferring random samples in actuator systems for binary damage detection," in *Proc. IEEE Int. Conf. Prognostics Health Manage.*, 2019, pp. 1–7.

[36] X. Xiao, D. Xu, and W. Wan, "Overview: Video recognition from handcrafted method to deep learning method," in *Proc. IEEE Int. Conf. Audio Lang. Image Process.*, 2016, pp. 646–651.

[37] L. Nanni, S. Ghidoni, and S. Brahnam, "Handcrafted vs. non-handcrafted features for computer vision classification," *Pattern Recognit.*, vol. 71, pp. 158–172, 2017.

[38] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[39] Z. Chen, T. Zhang, and C. Ouyang, "End-to-end airplane detection using transfer learning in remote sensing images," *Remote Sens.*, vol. 10, no. 1, pp. 139–153, 2018.

[40] S. Bulusu, B. Kailkhura, B. Li, P. Varshney, and D. Song, "Anomalous instance detection in deep learning: A survey," *IEEE Access*, vol. 8, pp. 132 330–132 347, 2020.

[41] F. A. Batarseh, L. Freeman, and C.-H. Huang, "A survey on artificial intelligence assurance," *J. Big Data*, vol. 8, no. 1, pp. 1–30, 2021.

[42] A. Holzinger, B. Malle, A. Saranti, and B. Pfeifer, "Towards multi-modal causability with graph neural networks enabling information fusion for explainable AI," *Inf. Fusion*, vol. 71, pp. 28–37, 2021.

[43] M. A. Hanif, F. Khalid, R. V. W. Putra, S. Rehman, and M. Shafique, "Robust machine learning systems: Reliability and security for deep neural networks," in *Proc. IEEE 24th Int. Symp. On-Line Testing Robust Syst. Des.*, 2018, pp. 257–260.

[44] G. Lecué and M. Lerasle, "Robust machine learning by median-of-means: Theory and practice," *Ann. Statist.*, vol. 48, no. 2, pp. 906–931, 2020.

[45] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018.

[46] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Proc. 23rd Nat. Conf. Artif. Intell.*, 2008, pp. 677–682.

[47] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2208–2217.

[48] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4068–4076.

[49] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, 2016.

[50] E. T. Jaynes, "Prior probabilities," *IEEE Trans. Syst. Sci. Cybern.*, vol. TSSC-4, no. 3, pp. 227–241, Sep. 1968.

[51] Y. LeCun, "The MNIST database of handwritten digits," 1998. [Online]. Available: http://yann. lecun. com/exdb/mnist/

[52] P. Kamsing, P. Torteeka, and S. Yooyen, "Deep convolutional neural networks for plane identification on satellite imagery by exploiting transfer learning with a different optimizer," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 9788–9791.

[53] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[54] M. Pal, "Random forest classifier for remote sensing classification," *Int. J. Remote Sens.*, vol. 26, no. 1, pp. 217–222, 2005.

[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[56] T. Cody, S. Adams, and P. A. Beling, "Empirically measuring transfer distance for system design and operation," 2021, *arXiv:2107.01184*.

Tyler Cody received the Ph.D. degree in systems engineering on a systems theory of transfer learning from the University of Virginia, Charlottesville, VA, USA, in 2021.

He is currently an Assistant Research Professor with the Virginia Tech National Security Institute, Blacksburg, VA. His research has been applied to machine learning for engineering systems broadly, including hydraulic actuators, industrial compressors, rotorcraft, telecommunication systems, and computer networks. His research interests include developing principles and best practices for the systems engineering of machine learning and artificial intelligence.

Stephen Adams (Associate Member, IEEE) received the M.S. degree in statistics and the Ph.D. degree in systems engineering from the University of Virginia, Charlottesville, VA, USA, in 2010 and 2015, respectively.

He is currently an Associate Research Professor with the Virginia Tech National Security Institute, Blacksburg, VA. He has experience developing and implementing numerous types of machine learning and artificial intelligence algorithms. His research has been applied to several domains including activity recognition, prognostics and health management, psychology, cybersecurity, data trustworthiness, natural language processing, and predictive modeling of destination given user geo-information data. His research interests include applications of machine learning, artificial intelligence in real-world systems, feature selection, machine learning with cost, transfer learning, reinforcement learning, and probabilistic modeling of systems.

Peter A. Beling received the Ph.D. degree in operations research from the University of California at Berkeley, Berkeley, CA, USA, in 1992.

He is currently a Professor with the Grado Department of Industrial and Systems Engineering, Blacksburg, VA, USA, and an Associate Director of the Intelligent Systems Laboratory, Virginia Tech National Security Institute, Blacksburg, VA. His research has found applications in a variety of domains, including mission engineering, cyber resilience of cyber-physical systems, prognostics and health management, and smart manufacturing. His research interests include the intersections of systems engineering and artificial intelligence (AI), and include AI adoption, reinforcement learning, transfer learning, and digital engineering.