A Systems Theory of Transfer Learning with Application

A

Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment

of the requirements for the degree

Doctor of Philosophy

by

Tyler Cody

May 2021

APPROVAL SHEET

This

Dissertation

is submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Author: Tyler Cody

Advisor: Peter Beling

Advisor:

Committee Member: Stephen Adams

Committee Member: Cody Fleming

Committee Member: William Scherer

Committee Member: Cong Shen

Committee Member:

Committee Member:

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, School of Engineering and Applied Science

A Systems Theory of Transfer Learning with Application

Tyler M. Cody

Abstract

Machine learning is an emerging technology with few principled engineering frameworks to guide its application. In particular, theoretical frameworks for understanding the interrelationships between systems and their learning processes are underdeveloped. The presented research addresses this gap by using Mesarovician abstract systems theory as a mathematical superstructure for learning theory, using the synthesized theory to characterize transfer learning systems, and operationalizing the resulting findings towards an empirical methodology for system design and operation. In particular, transfer distance, the abstract distance knowledge must traverse to be transferred from one system to another, is used as a metric for generalization difficulty, and thereby as a mechanism for relating the generalization of component learning systems to overall system design and operation. In sum, the presented research develops a systems theoretic framework for transfer learning and shows how it can be used to develop and organize best practices and tradecraft in systems engineering for artificial intelligence.

Acknowledgements

Mentorship has strongly shaped me. I thank Dr. Mildred Modugno for introducing me to the joy of problem solving. I thank Dr. Herbert Schilling for providing me the chance to rekindle that joy. And I thank Dr. Peter Beling for his many years of intentional mentorship and support.

I reserve a final note for my family. I thank my mother and father for their love and for all that they have instilled in me. And I thank my brothers for their commiseration and revelry.

Contents

1 Introduction **8**

2 Background **10**
2.1 General Systems Theory 10
2.2 Mesarovician Abstract Systems Theory 12
2.3 Transfer Learning 14
2.3.1 Generalization Bounds in Domain Adaptation 17

3 A Systems Theory of Transfer Learning **18**
3.1 Introduction 18
3.2 Learning Systems 19
3.3 Transfer Learning Systems 21
3.3.1 Comparison to Existing Frameworks 23
3.3.2 Transfer Approaches 24
3.3.3 Generalization in Transfer Learning 27
3.4 Structure and Behavior in Transfer Learning 30
3.4.1 Structural Considerations 31
3.4.2 Behavioral Considerations 34
3.4.3 Remarks 36
3.5 Conclusion 37

4 Transfer Distance for System Design and Operation **38**
4.1 Introduction 38
4.2 Background 39
4.2.1 Concept Drift 39
4.2.2 Prognostics and Health Management 40
4.2.3 Computer Vision 41
4.3 Methods 41
4.4 Transfer Distance for System Design 43
4.4.1 Transfer Distance Induced by Rebuild 44
4.4.2 Transfer Distance and Sample Size 47
4.5 Transfer Distance for System Operation 49
4.5.1 MNIST and Expected Operational Performance 50
4.5.2 Mission Scenario in Aircraft Detection 53
4.6 Conclusion 55

5 Closing Remarks **57**

6 Appendix **59**
6.1 Mesarovician Glossary 59
6.2 Learning Systems 60

Notation

Let X denote a set and $x \in X$ denote its elements. For notational convenience random variables are not distinguished—probability measures on X are denoted $P(X)$. The Cartesian product is denoted $\times$, and for any object $V_i = V_{i1} \times \ldots \times V_{in}$, $\overline{V_i}$ shall denote the family of component sets of V_i, $\overline{V_i} = \{V_{i1}, \ldots, V_{in}\}$. The cardinality of X is denoted $|X|$. The powerset is denoted $\mathcal{P}$. Herein it has two uses. Frequently, in order to express input-output conditions for a learning system we will only use its input-output representation $S : D \times X \to Y$. In contexts where $S \subset \times\{A, D, \Theta, H, X, Y\}$, we use $(d, x, y) \in \mathcal{P}(S)$ to make reference to the input-output representation. Also, the subset of the powerset of a powerset $K \subset \mathcal{P}(\mathcal{P}(D \cup \Theta))$ is used to denote that K can be $\subset D$, $\subset \Theta$, or $\subset D \times \Theta$, etc., i.e., to make reference to ordered pairs. Often, we make reference to $d \in D$ to say a particular set of data d from the larger set D. Additionally, for a system $S \subset X \times Y$, when we discuss $x \in X$ or $y \in Y$ it is assumed that $(x, y) \in S$ unless stated otherwise. This is to save the reader from the pedantry of Mesarovician abstract systems theory.

1 Introduction

Machine learning has moved beyond being a research field to being a workable approach for building autonomous functions into systems, however, a principled discipline for the systems engineering of systems with learning algorithms has yet to emerge. Although learning algorithms offer statistical improvements in performance and allow for novel functionality, their interactions with the systems within which they are embedded raise concerns.

Frameworks for studying these relationships are underdeveloped. Although research in cybernetics and general systems theory has explored these issues for decades [24], the bulk of its findings rely on metaphorical abstractions that often abstract the frameworks away from the specific nature of the underlying learning processes. Conversely, machine learning and learning theoretic approaches focus on the learning processes and learning algorithms in isolation [58, 87], thereby lacking the broader systems context and neglecting the broader systems view.

From this observation, it appears desirable to find a middle ground. By sacrificing some of the generality of general systems theory and some of the specificity of learning theory, a Mesarovician abstract systems theory (AST) of learning is able to closely knit systems and learning theory together. AST is a mathematical framework for studying the nature of general systems [56]. By using systems theory as a superstructure for learning, learning algorithms can be formally studied in the context of the systems within which they operate.

Formal systems theory is often seen as a conceptual tool and an unnecessary cost, if not a hindrance, in detailed modeling and analysis. However, since learning theory and machine learning are largely mathematical constructs, mathematical systems theory offers an appropriate meta theory, and, further, since AST and learning theory are similarly general in their pursuits of general understandings, their respective understandings ought to be readily expressible in each others' terms. This observation bears out mathematically. Systems theory is largely a theory of sets. Learning theory is largely a theory of probability, or, in other words, a theory of measures on those sets. Because of this closeness, significant general systems elaboration on learning can be afforded without a loss of parsimony, and general systems results can be translated into the terms of learning theory or those of particular solutions methods of interest.

Here, the focus centers on transfer learning. Transfer learning describes the ability of a system to use knowledge learned in previous tasks to help learn novel tasks. Instead of studying transfer learning as a learning algorithm, as in machine learning, or as a learning process, as in learning theory, we formalize it as a system. By elaborating on this formalization, we arrive at formal, general systems notions of transferrability, transfer roughness, and transfer distance. Transfer dis-

tance describes the abstract distance knowledge must traverse to be transferred from one system to another. Informed by our characterization of transfer learning as a system, we demonstrate transfer distance's use as a metric for systems engineering, particularly for system design and operation.

Systems theory can serve as a foundation for a principled discipline of systems engineering for artificial intelligence (AI) by providing a framework for developing and organizing best practices and tradecraft. In transfer learning, such a systems theoretic framework, as presented herein, stands in stark contrast to existing machine learning frameworks with respect to the breadth of its formalism and its overall perspective. By taking a top-down, systems approach to formalizing transfer learning, we arrive at a characterization of transfer learning as a general, mathematical construct without explicit reference to solution methods. Moreover, the framework can integrate those aspects of learning theory and machine learning pertinent to a system of interest without restructuring the framework. As such, the framework can be used both to make general considerations about best practices in transfer learning systems and also to arrive at discipline-specific tradecraft for realizing those systems.

The structure of this dissertation is as follows. First, background on systems theory and transfer learning is given in Section 2. The body of the dissertation is divided into two parts. Section 3 presents the systems theoretical framework for transfer learning systems, Section 4 demonstrates its use in developing and organizing best practices and tradecraft for system design and operation, and together they show that systems theory can be used as a foundation for a principled discipline of systems engineering for AI. Section 5 concludes the dissertation. Supplementary material can be found in the succeeding Appendix.

2 Background

In the following we review general systems theory and introduce pertinent Mesarovician abstract systems theory. Then we review transfer learning and make explicit the principal differences between existing frameworks and ours. These preliminaries support the dissertation at large. Specific background applying to Section 4 is given therein. A supplemental glossary of Mesarovician terms can be found in the Appendix.

2.1 General Systems Theory

Set-theoretic systems theory draws from two principal schools of thought: Ludwig von Bertalanffy's general systems theory [8,9,90] and Norbert Weiner's cybernetics [50,71,94]. Many scientists and researchers have contributed to the schools, notably Anatol Rapoport [68, 69] and Kenneth Boulding [11] to von Bertalanffy's and Ross Ashby [4] and Herbert Simon [78, 79] to Weiner's. The schools experienced extensive co-development and it is difficult to dichotomize contributing authors.

Ludwig von Bertalanffy expresses his vision for general systems theory articulately using an observation [90]:

> "... there exist models, principals, and laws that apply to generalized systems or their subclasses, irrespective of their particular kind, the nature of their component elements, and the relationships or "forces" between them. It seems legitimate to ask for a theory, not of systems of a more or less special kind, but of universal principles applying to systems in general."

Despite his intentions, his work and those closely following his school of thought fell short of developing a rigorous theory. Their work can be characterized as a descriptive approach to general systems theory that makes a heavy reliance on metaphors instead of axioms.

Cybernetics was more mathematically grounded than von Bertalanffy's general systems theory at its conception [50,71,94]. In the field's seminal text [94], Weiner defines cybernetics as:

> "the scientific study of control and communication in the animal and the machine."

Importantly, his work and that of others in the field showed that interdisciplinary problems can be treated mathematically, and that control processes can be found everywhere in nature.

Drawing from the success of formalism in cybernetics and the vision of general systems theory, subsequent work by Jay Forrester [28], George Klir [38], A. Wayne Wymore [98], and M.D. Mesarovic [53, 55], among others, laid the foundations for formal general systems theories. Wymore and Mesarovic, in contrast to many of their contemporaries, tried to directly formulate a mathematical general systems theory. Both Wymore and Mesarovic took set-theoretic approaches, however, whereas the formalism of Wymore's mathematical structure was heavier and biased towards engineering applications [96,97], Mesarovic's was light-weight and highly abstract.

Mesarovician systems theory, referred to by its originator first as *general systems theory* [55], and later by the more distinguishing title *abstract systems theory* [56], is a set-theoretic mathematical framework that seeks to realize von Bertalanffy's vision in a way that is "simple, elegant, general, and precise" [55]. Concepts are introduced axiomatically, and mathematical structures needed to do so are introduced such that the formalisms are precise without losing their generality. In arguing for such a mathematical approach, Mesarovic states that [55]:

> "...the investigation of the logical consequences of systems having given properties should be of central concern for any general systems theory which cannot be limited solely to a descriptive classification of systems."

Mesarovic develops his theory using a process he refers to as *formalization*. The process involves giving a verbal description a precise mathematical definition using as few axioms as possible. Mathematical structure is added as needed to specify systems properties of interest. Thus, the formalization approach to general systems theory naturally identifies how fundamental particular systems properties are relative to others. Mesarovician systems theorists use the formalization approach to specify properties and arrive at specific classes of systems. The relationships between classes of systems can then be formally studied using a category theory of systems [56]. What results is a mathematically explicit understanding of how very general classes of systems relate to each other.

Mesarovic's systems theory was extended throughout the 20th century, notably by Yasuhiko Takahara and Donald Macko [49,56,57]. Mesarovic and others applied his framework to biological systems [52, 54], however, most results strictly following his initial theory were purely mathematical. The work presented herein extends Mesarovic's framework for studying general systems to study general learning processes in general systems, thereby helping to fill the gap in formal frameworks for characterizing learning in a systems context. Specifically, we first formalize learning systems, then formalize transfer learning as a relation on learning systems, and finally connect the formalism to practice by using it to inform empirical methodology for system design and operation.

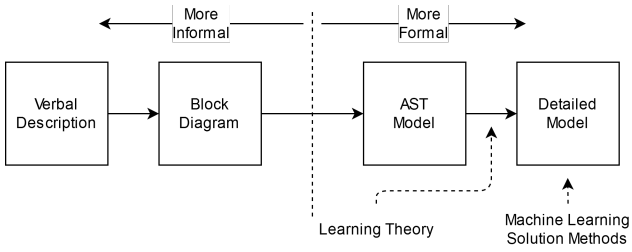


Figure 1: AST is a minimally formal framework. In modeling learning, learning theory brings formalism to AST, and machine learning specifies the detailed model.

2.2 Mesarovician Abstract Systems Theory

Mesarovician abstract systems theory is a general systems theory that adopts the formal minimalist world-view [24, 56]. AST is developed top-down, with the goal of giving a verbal description a parsimonious yet precise mathematical definition. Mathematical structure is added as needed to specify systems properties of interest. This facilitates working at multiple levels of abstraction within the same framework, where mathematical specifications can be added without restructuring the framework. In modeling, it is used as an intermediate step between informal reasoning and detailed mathematics by formalizing block-diagrams with little to no loss of generality, see Figure 1. Apparently this generality limits its deductive powers, but, in return, it helps uncover fundamental mathematical structure related to the general characterization and categorization of phenomena.

We will now review the AST definitions of a system, input-output system, and goal-seeking system, and the related notions of system structure and behavior. Additional details can be found in the Appendix.

In AST, a system is defined as a relation on component sets. When those sets can be partitioned, the system is called an input-output system. Systems and input-output systems are defined as follows.

Definition 1. System.
A (general) system is a relation on non-empty (abstract) sets,

$$S \subset \times\{V_i : i \in I\}$$

where $\times$ denotes the Cartesian product and I is the index set. A component set V_i is referred to as a system object.

Definition 2. Input-Output Systems.
Consider a system S, where $S \subset \times\{V_i : i \in I\}$. Let $I_x \subset I$ and $I_y \subset I$ be a partition of I, i.e., $I_x \cap I_y = \emptyset$, $I_x \cup I_y = I$. The set $X = \times\{V_i : i \in I_x\}$ is termed

the input object and $Y = \times\{V_i : i \in I_y\}$ *is termed the output object. The system is then*

$$S \subset X \times Y$$

and is referred to as an input-output system. If S *is a function* $S : X \to Y$*, it is referred to as a function-type system.*

AST is developed by adding structure to the component sets and the relation among them. Input-output systems with an internal feedback mechanism are referred to as goal-seeking (or cybernetic) systems. The internal feedback of goal-seeking systems is specified by a pair of consistency relations G and E which formalize the notions of goal and seeking, respectively. Figure 2 depicts input-output and goal-seeking systems. Goal-seeking systems are defined as follows.

Definition 3. Goal-Seeking Systems.
A system $S : X \to Y$ *has a goal-seeking representation if there exists a pair of maps*

$$S_G : X \times Y \to \Theta$$
$$S_F : \Theta \times X \to Y$$

and another pair

$$G : \Theta \times X \times Y \to V$$
$$E : X \times Y \times V \to \Theta$$

such that

$$(x, y) \in S \leftrightarrow (\exists\theta)[(\theta, x, y) \in S_F \wedge (x, y, \theta) \in S_G]$$
$$(x, y, G(\theta, x, y), \theta) \in E \leftrightarrow (x, y, \theta) \in S_G$$

where

$$x \in X, y \in Y, \theta \in \Theta.$$

S_G *is termed the goal-seeking system and* S_F *the functional system.* G *and* E *are termed the goal and seeking relations, and* V *the value.*

System structure and behavior are focal in Mesarovician characterizations of systems. System structure refers to the mathematical structure of a system's component sets and the relations among them. For example, there may be algebraic structure related to the specification of the relation, e.g. the linearity of a relationship between two component sets. System behaviors, in contrast, are properties or descriptions paired with systems. For example, consider a system $S : X \to Y$

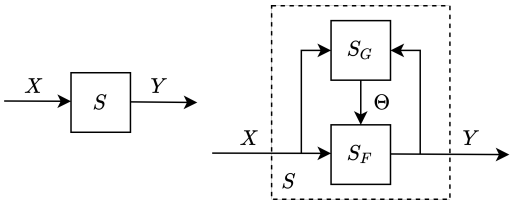


Figure 2: Input-output systems (left) and goal-seeking systems (right).

and a map $S \rightarrow \{stable, neutral, unstable\}$. A linear increasing function and an increasing power function may both be considered behaviorally unstable, but clearly their structures are different [56].

Similarity of systems is a fundamental notion, and it can be expressed well in structural and behavioral terms. Structural similarity describes the *homomorphism* between two systems' structures. Herein, in accord with category theory, a map from one system to another is termed a morphism, and homomorphism specifies the morphism to be onto. Homomorphism is formally defined as follows.

Definition 4. Homomorphism.
An input-output system $S \subset X \times Y$ is homomorphic to $S' \subset \times X' \times Y'$ if there exists a pair of maps,

$$\varrho : X \rightarrow X', \vartheta : Y \rightarrow Y'$$

such that for all $x \in X$, $x' \in X'$, and $y \in Y$, $y' \in Y'$, $\varrho(x) = x'$ and $\vartheta(y) = y'$.

Behavioral similarity, in contrast, describes the *proximity* or *distance* between two systems' behavior. As in AST generally, we use structure and behavior as the primary apparatus for elaborating on our formulation of transfer learning systems. Refer to the Appendix for additional details on structure, behavior, and similarity.

2.3 Transfer Learning

DARPA describes transfer learning as "the ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks" in Broad Agency Announcement (BAA) 05-29. The previous tasks are referred to as source tasks and the novel task is referred to as the target task. Thus, transfer learning seeks to transfer knowledge from some source learning systems to a target learning system.

Transfer learning is widely studied by computer scientists [19,65,73,83,93,101]. The generality of the transfer learning framework makes it a super-structure for many learning problems. It is closely related to other generalization mechanisms such as multi-task learning [105], domain adaptation [6, 66], and concept drift

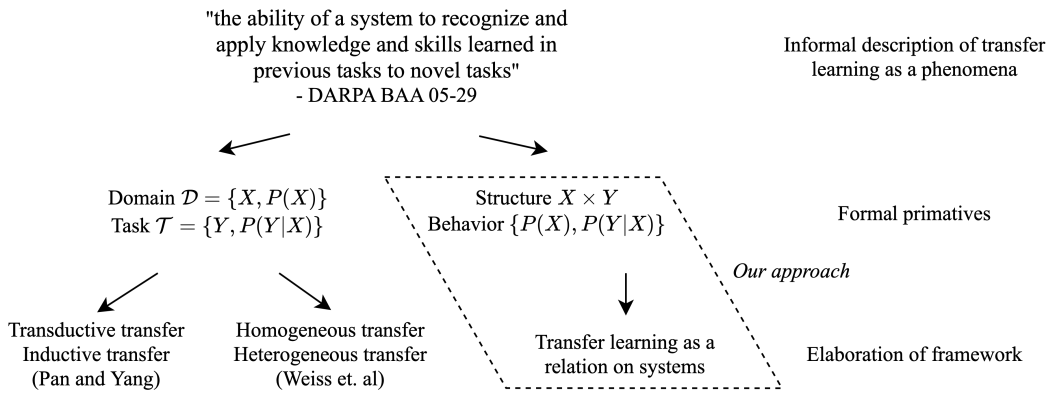


Figure 3: Existing frameworks interpret the informal definition of transfer learning given by DARPA in terms of domain $\mathcal{D}$ and task $\mathcal{T}$. In contrast, we use structure and behavior, which provide a more formal basis for elaboration.

[23, 29, 36, 92, 106, 107], which focus on sharing data, features, and parameters to improve learning, and loosely related to meta-learning and learning to learn [43, 44, 81, 88, 89], which focus on using meta-data about the learning process itself to improve learning.

Transfer learning enables learning in environments where data is 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 X and labels Y [16]. 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 [64].

Identifying whether or not transfer learning is an appropriate solution for a particular learning problem is crucial [72]. 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 [91]. Thus, closeness between the source and target distributions is a pre-condition for transfer learning success.

Existing frameworks for transfer learning focus on a dichotomy between *domain* $\mathcal{D}$ and *task* $\mathcal{T}$. The domain $\mathcal{D}$ consists of the input space X and its marginal distribution $P(X)$. The task $\mathcal{T}$ consists of the output space Y and its posterior distribution $P(Y|X)$. The seminal transfer learning survey frames transfer learning

in terms of an inequality of domains $\mathcal{D}$ and tasks $\mathcal{T}$ [64]. Therein, Pan and Yang define transfer learning as follows.

Definition 5. Transfer learning.
Given a source domain $\mathcal{D}_S$ and task $\mathcal{T}_S$ and a target domain $\mathcal{D}_T$ and task $\mathcal{T}_T$, transfer learning aims to improve the learning of $P(Y_T|X_T)$ in the target using knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$.

Pan and Yang continue by defining *inductive transfer* as the case where the source and target tasks are not equal, $\mathcal{T}_S \neq \mathcal{T}_T$, and *transductive transfer* as the case where the source and target domains are not equal but their tasks are, $\mathcal{D}_S \neq \mathcal{D}_T \wedge \mathcal{T}_S = \mathcal{T}_T$. They use these two notions, and their sub-classes, to categorize the transfer learning literature and its affinity for related fields of study. Alternative frameworks use notions of *homogeneous* and *heterogeneous* transfer, which correspond to the cases where the sample spaces of the source and target domains X and tasks Y are or are not equal, respectively [93].

While these formalisms describe the literature well, they are not rich enough to maintain formalism in the elaboration of their respective frameworks. For example, Pan and Yang address what, how, and when to transfer in a largely informal manner, making reference to inductive and transductive transfer as guideposts, but ultimately resorting to verbal descriptions [64]. In contrast, instead of starting with domain $\mathcal{D}$ and task $\mathcal{T}$ as the fundamental notions of transfer learning, we use structure and behavior—two concepts with deep general systems meaning, define transfer learning as a relation on systems, and carry formalism through into subsequent elaboration. The principal difference between existing frameworks and ours is depicted in Figure 3. Importantly, despite our formalism, we maintain a general systems level of abstraction, in contrast to purely learning theoretical frameworks for transfer learning [40]. As such, we compare our general framework with those of Pan and Yang [64] and Weiss et. al [93]. The presented work greatly expands on previous, initial efforts in this direction [15, 16].

An important and under-studied related notion in transfer learning is *transfer distance*. Transfer distance is an informal term used to describe the abstract distance between learning tasks. Accordingly, far transfer refers to dissimilar tasks, and near transfer refers to similar tasks. While many transfer learning algorithms involve computing a transfer distance, e.g., maximum mean discrepancy [62], as a distributional-divergence-based component of loss functions, there are few works studying it directly [5, 10, 59, 102], let alone studying its use in engineering practice. The presented research helps fill this gap by formalizing transfer distance in systems theoretic terms, wherein transfer distance is a function of the behavioral similarity of learning systems, and applying this formalism towards empirical methodology for system design and operation.

2.3.1 Generalization Bounds in Domain Adaptation

Domain adaptation is a sub-field of transfer learning where $X_S \times Y_S = X_T \times Y_T$ [33]. 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 [5, 10]. 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, δ is the transfer distance, and C is a constant term which accounts for relevant complexities, for example, VC-dimension [5]. Although Inequality 1 is an approximation of the underlying learning theory, specifications can be added to arrive at proper, learning theoretic bounds using statistical divergence [10], $\mathcal{H}$-divergence [5], Rademacher complexity [59], or integral probability metrics [102].

3 A Systems Theory of Transfer Learning

3.1 Introduction

Transfer learning, unlike classical learning, does not assume that the training and operating environments are the same, and, as such, is fundamental to the development of real-world learning systems. In transfer learning, knowledge from various *source* sample spaces and associated probability distributions is *transferred* to a particular *target* sample space and probability distribution. Transfer learning enables learning in environments where data is limited. Perhaps more importantly, it allows learning systems to propagate their knowledge forward through distributional changes.

Mechanisms for knowledge transfer are a bottleneck in the deployment of learning systems. Learning in identically distributed settings has been the focus of learning theory and machine learning research for decades, however, such settings represent a minority of use cases. In real-world settings, distributions and sample spaces vary between systems and evolve over time. Transfer learning addresses such differences by sharing knowledge between learning systems, thus offering a theory principally based on distributional difference, and thereby a path towards the majority of use cases.

Existing transfer learning frameworks are incomplete from a systems theoretic perspective. They focus on domain and task, and neglect perspectives offered by explicitly considering system structure and behavior. Mesarovician systems theory can be used as a super-structure for learning to top-down model transfer learning, and although existing transfer learning frameworks may better reflect and classify the literature, the resulting systems theoretic framework offers a more rigorous foundation better suited for system design and analysis.

Mesarovician systems theory is a set-theoretic meta-theory concerned with the characterization and categorization of systems. A system is defined as a relation on sets and mathematical structure is sequentially added to those sets, their elements, or the relation among them to formalize phenomena of interest. By taking a top-down, systems approach to framing transfer learning, instead of using a bottom-up survey of the field, we naturally arrive at a framework for modeling transfer learning without necessarily referencing solution methods. This allows for general considerations of transfer learning systems, and is fundamental to the understanding of transfer learning as a mathematical construct.

We provide a novel definition of transfer learning systems, dichotomize transfer learning in terms of structure and behavior, and formalize notions of negative transfer, transferability, transfer distance, and transfer roughness in subsequent elaborations. This section is structured as follows. First we define learning systems and discuss their relationship to abstract systems theory and empirical risk minimization

in Section 3.2. Using this definition, transfer learning systems are defined and studied in Sections 3.3 and 3.4. We conclude with a synopsis and remarks in Section 3.5.

3.2 Learning Systems

We follow Mesarovic's top-down process to sequentially construct a learning system S. Learning is a relation on data and hypotheses. To the extent that a scientific approach is taken, those hypotheses are explanations of initial-final condition pairs [67]. Otherwise put, we are concerned with learning as function estimation. We additionally note that learning algorithms use data to select those hypotheses and that the data is a sample of input-output pairs [87]. Such a learning system can be formally defined as follows.

Definition 6. (Input-Output) Learning System.
A learning system S is a relation

$$S \subset \times\{A, D, \Theta, H, X, Y\}$$

such that

$$D \subset X \times Y, A : D \to \Theta, H : \Theta \times X \to Y$$
$$(d, x, y) \in \mathcal{P}(S) \leftrightarrow (\exists\theta)[(\theta, x, y) \in H \wedge (d, \theta) \in A]$$

where

$$x \in X, y \in Y, d \in D, \theta \in \Theta.$$

The algorithm A, data D, parameters Θ, hypotheses H, input X, and output Y are the component sets of S, and learning is specified in the relation among them.

The above definition of learning formalizes learning as a cascade connection of two input-output systems: an inductive system $S_I \subset \times\{A, D, \Theta\}$ responsible for inducing hypotheses from data, and a functional system $S_F \subset \times\{\Theta, H, X, Y\}$, i.e. the induced hypothesis. S_I and S_F are coupled by the parameter Θ. Learning is hardly a purely input-output process, however. To address this, we must specify the goal-seeking nature of S_I, and, more particularly, of A.

A is goal-seeking in that it makes use of a *goal* relation $G : D \times \Theta \to V$ that assigns a value $v \in V$ to data-parameter pairs, and a *seeking* relation $E : V \times D \to \Theta$ that assigns parameter $\theta \in \Theta$ to data-value pairs. These consistency relations G and E specify A, but not by decomposition; i.e., in general, G and E cannot be composed to form A. The definition of a learning system can be extended as follows.

Definition 7. (Goal-Seeking) Learning System.
A learning system S is a relation

$$S \subset \times\{A, D, \Theta, G, E, H, X, Y\}$$

such that

$$D \subset X \times Y, A : D \to \Theta, H : \Theta \times X \to Y$$
$$(d, x, y) \in \mathcal{P}(S) \leftrightarrow (\exists\theta)[(\theta, x, y) \in H \wedge (d, \theta) \in A]$$

and

$$G : D \times \Theta \to V, E : V \times D \to \Theta$$
$$(d, G(\theta, d), \theta) \in E \leftrightarrow (d, \theta) \in A$$

where

$$x \in X, y \in Y, d \in D, \theta \in \Theta.$$

The algorithm A, data D, parameters Θ, consistency relations G and E, hypotheses H, input X, and output Y are the component sets of S, and learning is specified in the relation among them.

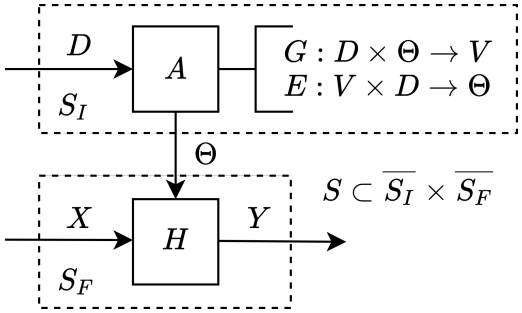


Figure 4: Learning systems are a cascade connection of the inductive system S_I and the induced hypothesis S_F. S_I is goal-seeking.

Learning systems are depicted in Figure 4. These systems theoretic definitions of learning have an affinity to learning theoretic constructions. Consider empirical risk minimization (ERM), where empirical measures of risk are minimized to determine the optimal hypothesis for a given sample [87]. Apparently, ERM specifies G to be a measure of risk calculated on the basis of a sample drawn independently according to a probability measure on the approximated function $f : X \to Y$ and specifies E to be a minimization of G over Θ.

We have demonstrated how our definition of a learning system anchors our framework to both AST and ERM. We posit these definitions not as universal truths, but rather as constructions that anchor our framing of transfer learning to systems and learning theory. We abstain from further elaboration on these definitions, however, proofs of the above propositions can be found in the Appendix. In the following, we leave G and E implicit, only making reference to f and related probability measures.

Example 3.1. *Learning in an Unmanned Aerial Vehicle.*
Consider an unmanned aerial vehicle (UAV) with a learning system S for path planning. H is a function from sensor data X, e.g., from accelerometers, cameras, and radar, to flight paths Y. D, then, consists of sets of sensor-path pairs. If S is a support-vector machine (SVM), then H is a set of half-spaces parameterized by Θ and A is a convex optimization routine [82]. The inductive system S_I consists of the optimization routine A and is responsible for selecting path planning models $h \in H$. Those models h form the functional system S_F which takes in sensor data X and outputs paths Y.

3.3 Transfer Learning Systems

Transfer learning is conventionally framed as a problem of sharing knowledge from source domains and tasks to a target domain and task. We propose an alternative approach. We formulate transfer learning top-down in reference to the source and target learning systems, and then dichotomize subsequent analysis not by domain and task, but rather by structure, described primarily by the $X \times Y$ space, and behavior, described primarily by probability measures on the estimated function $f : X \to Y$.

A transfer learning system is a relation on the source and target systems that combines knowledge from the source with data from the target and uses the result to select a hypothesis that estimates the target learning task f_T. We define it formally as follows.

Definition 8. Transfer Learning System.
Given source and target learning systems S_S and S_T

$$S_S \subset \times\{A_S, D_S, \Theta_S, H_S, X_S, Y_S\}$$
$$S_T \subset \times\{A_T, D_T, \Theta_T, H_T, X_T, Y_T\}$$

a transfer learning system S_{Tr} is a relation on the component sets of the source and target systems

$$S_{Tr} \subset \overline{S_S} \times \overline{S_T}$$

such that

$$K_S \subset D_S \times \Theta_S, D \subset D_T \times K_S$$

and

$$A_{Tr} : D \to \Theta_{Tr}, H_{Tr} : \Theta_{Tr} \times X_T \to Y_T$$
$$(d, x_T, y_T) \in \mathcal{P}(S_{Tr}) \leftrightarrow (\exists \theta_{Tr})[(\theta_{Tr}, x_T, y_T) \in H_{Tr} \wedge (d, \theta_{Tr}) \in A_{Tr}]$$

where

$$x_T \in X_T, y_T \in Y_T, d \in D, \theta_{Tr} \in \Theta_{Tr}.$$

The nature of source knowledge K_S[1], the transfer learning algorithm A_{Tr}, hypotheses H_{Tr}, and parameters Θ_{Tr} specify transfer learning as a relation on $\overline{S_S}$ and $\overline{S_T}$.

Trivial transfer occurs when the structure and behavior of S_S and S_T are the same, or, otherwise put, when transfer learning reduces to classical, identically distributed learning. Transfer is non-trivial when there is a structural difference $X_S \times Y_S \neq X_T \times Y_T$ or a behavioral difference $P(X_S) \neq P(X_T) \vee P(Y_S|X_S) \neq P(Y_T|X_T)$ between the source S_S and target S_T. If the posterior distributions $P(Y|X)$ and marginal distributions $P(X)$ are equal between the source and target systems, then transfer is trivial. Non-trivial transfer is implied when $X_S \times Y_S \neq X_T \times Y_T$.

Proposition. S_{Tr} in Definition 8 is a learning system as defined in Definition 6.
Proof: *As stated in Definition 8, a transfer learning system is a relation $S_{Tr} \subset \overline{S_S} \times \overline{S_T}$. More particularly, it is a relation $S_{Tr} \subset (D_S \times \Theta_S) \times (D_T \times X_T \times Y_T)$, and has a function-type representation $S_{Tr} : D_S \times \Theta_S \times D_T \times X_T \to Y_T$. Its inductive system is the relation $A_{Tr} : D \to \Theta_{Tr}$, where $D \subset D_S \times \Theta_S \times D_T$. And its functional system is the relation $H_{Tr} : \Theta_{Tr} \times X_T \to Y_T$. Thus, we can restate S_{Tr} as a relation*

$$S_{Tr} \subset \times\{A_{Tr}, D, \Theta_{Tr}, H_{Tr}, X_T, Y_T\}$$

and since by Definition 8

$$(d, x_T, y_T) \in \mathcal{P}(S_{Tr}) \leftrightarrow$$
$$(\exists \theta_{Tr})[(\theta_{Tr}, x_T, y_T) \in H_{Tr} \wedge (d, \theta_{Tr}) \in A_{Tr}]$$

where

$$x_T \in X_T, y_T \in Y_T, d \in D, \theta_{Tr} \in \Theta_{Tr},$$

we have that S_{Tr} is an input-output learning system as in Definition 6.

[1]Here, we define the transferred knowledge K_S to be D_S and Θ_S, the source data and parameters, following convention [64]. In general, however, source knowledge $K_S \subset \mathcal{P}(\mathcal{P}(\overline{S_S}))$.

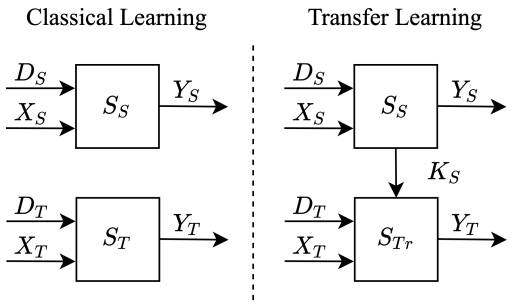


Figure 5: Transfer learning systems S_{Tr} are a relation $K_S \times D_T \times X_T \to Y_T$, while the target system S_T is a relation $D_T \times X_T \to Y_T$.

Transfer learning systems are distinguished from general learning systems by the selection and transfer of K_S, and its relation to D_T by way of $D \subset K_S \times D_T$ and its associated operator $K_S \times D_T \to D$. In cases where $\{A_{Tr}, \Theta_{Tr}, H_{Tr}\} \leftrightarrow \{A_T, \Theta_T, H_T\}$, e.g., as is possible when transfer learning consists of pooling samples with identical supports, the additional input K_S is all that distinguishes S_{Tr} from S_T. Classical and transfer learning systems are depicted in Figure 5.

As we will see, however, this is no small distinction, as it allows for consideration of learning across differing system structures and behaviors. But before we elaborate on the richness of structural and behavioral considerations, first, in the following subsections, we interpret existing frameworks in terms of structure and behavior and define preliminary notions related to generalization in transfer learning.

Example 3.2. *Transfer Learning in UAVs.*
Consider UAVs with learning systems S_S and S_T defined according to Example 3.1 and a transfer learning system $S_{Tr} \subset \overline{S_S} \times \overline{S_T}$. If S_{Tr} is also a SVM, then H_{Tr} are also half-spaces parameterized by Θ_{Tr}. If $K_S \subset D_S \times \Theta_S$, Θ_S can provide an initial estimate for Θ_{Tr}, and D_S can be pooled with D_T to update this estimate. A_{Tr}, in distinction to A_T, must facilitate this initialization and pooling.

3.3.1 Comparison to Existing Frameworks

Using Definition 8, the central notions of existing frameworks can be immediately defined in terms of structural and behavioral inequalities. Homogeneous transfer specifies structural equality of the source and target sample spaces, $X_S \times Y_S = X_T \times Y_T$, and heterogeneous transfer specifies otherwise. Domain adaptation, covariate shift, and prior shift are all examples of homogeneous transfer [18, 33, 64]. Transductive and inductive transfer entail more nuanced specifications.

Recall, inductive transfer specifies that $\mathcal{T}_S \neq \mathcal{T}_T$ and transductive transfer specifies that $\mathcal{D}_S \neq \mathcal{D}_T \wedge \mathcal{T}_S = \mathcal{T}_T$, where $\mathcal{D} = \{P(X), X\}$ and $\mathcal{T} = \{P(Y|X), Y\}$.

Technically, transductive transfer occurs if $X_S \neq X_T$ or if $P(X_S) \neq P(X_T)$. However, if $X_S \neq X_T$, then it is common for $P(Y_S|X_S) \neq P(Y_T|X_T)$ because the input set conditioning the posterior has changed, and thus it is likely that $\mathcal{T}_S \neq \mathcal{T}_T$. To that extent, in the main, transductive transfer specifies a difference between input behavior while output behavior remains equal. Inductive transfer, on the other hand, is more vague, and merely specifies that there is a structural difference in the outputs, $Y_S \neq Y_T$, or a behavioral difference in the posteriors, $P(Y_S|X_S) \neq P(Y_S|X_T)$. Note, this behavioral difference in the posteriors can be induced by a structural difference in the inputs as previously mentioned, and is implied by a structural difference in the outputs.

In short, the homogeneous-heterogeneous dichotomy neglects behavior and the transductive-inductive framing muddles the distinction between structure and behavior. While frameworks based on either cover the literature well, they only provide high-level formalisms which are difficult to carry through into general, formal characterizations of transfer learning systems. In contrast, Definition 8 provides a formalism that can be used to define transfer learning approaches and auxiliary topics in generalization.

3.3.2 Transfer Approaches

Consider how the seminal framework informally classifies transfer learning algorithms [64]. Three main approaches are identified: 'instance transfer', 'parameter transfer', and 'feature-representation transfer'. While the transductive or inductive nature of a transfer learning system gives insight into which approaches are available, the approaches cannot be formalized in those terms, or in terms of domain $\mathcal{D}$ and task $\mathcal{T}$ for that matter, because they are a specification on the inductive system $S_I \subset \times\{A_{Tr}, D_{Tr}, \Theta_{Tr}\}$, whereas the former are specifications on the functional system $S_F \subset \times\{\Theta_{Tr}, H_{Tr}, X_{Tr}, Y_{Tr}\}$.

With the additional formalism of Definition 8, these transfer approaches can be formalized using system structure. First, note that differently structured data D leads to different approaches. Consider the categories of transfer learning systems corresponding to the various cases where $D \subset \mathcal{P}(\mathcal{P}(D_T \cup D_S \cup \Theta_S))$. Instance and parameter transfer correspond to transferring knowledge in terms of D_S and Θ_S, respectively, and can be formally defined as follows.

Definition 9. Instance Transfer.
A transfer learning system S_{Tr} is an instance transfer learning system if $K_S \subset D_S$, i.e., if

$$\mathcal{A}_{Tr} : D \to \Theta_{Tr} \iff \mathcal{A}_{Tr} : D_S \times D_T \to \Theta_{Tr}.$$

Definition 10. Parameter Transfer.
A transfer learning system S_{Tr} is a parameter transfer learning system if $K_S \subset \Theta_S$, i.e., if

$$\mathcal{A}_{Tr} : D \to \Theta_{Tr} \iff \mathcal{A}_{Tr} : \Theta_S \times D_T \to \Theta_{Tr}.$$

Feature-representation transfer, in contrast, specifies that learning involves transformations on $\overline{S_T}$, K_S, or both. It can be defined formally as follows.

Definition 11. Feature-Representation Transfer.
Consider a transfer learning system S_{Tr} and a learning system S_L, termed the latent learning system. Note, S_{Tr} and S_L can be represented as function-type systems,

$$S_{Tr} : D \times X_T \to Y_T$$
$$S_L : D_L \times X_L \to Y_L.$$

S_{Tr} is a feature-representation transfer learning system if there exist maps

$$m_D : D \to D_L, m_{X_T} : X_T \to X_L, m_{Y_L} : Y_L \to Y_T$$

such that

$$\forall (d, x_T, y_T) \in (S_{Tr})$$
$$S_{Tr}(d, x_T) \leftrightarrow m_{Y_L}(S_L(m_D(d), m_{X_T}(x_T)))$$

where

$$d \in D, x_T \in X_T, y_T \in Y_T.$$

In other words, S_{Tr} is a feature-representation transfer learning system if transfer learning involves transforming to and from a latent system where learning occurs.

Proposition. Learning in S_S, S_T, and S_L.
Consider a case of feature-representation transfer where $K_S \subset D_S$. Let $m_{D_T} : D_T \to D_L$ and $m_{D_S} : D_S \to D_L$. Then, $m_D \iff (m_{D_T}, m_{D_S})$. Recall $D_i \subset X_i \times Y_i$. If m_{D_T} is the identity and m_{D_S} is not, then $X_T \times Y_T = X_L \times Y_L$—learning occurs in the target sample space. If m_{D_S} is the identity and m_{D_T} is not, then $X_S \times Y_S = X_L \times Y_L$—learning occurs in the source sample space. If m_D is the identity, then $X_S \times Y_S = X_T \times Y_T = X_L \times Y_L$, i.e., S_{Tr} involves homogeneous transfer. If neither m_{D_T} or m_{D_S} are the identity, then learning occurs in a latent sample space $X_L \times Y_L$ that is unequal to $X_T \times Y_T$ and $X_S \times Y_S$.

In feature-representation transfer, data $D \subset D_T \times K_S$ is mapped to a latent system S_L where learning occurs. By way of $m_D : D \to D_L$, feature-representation transfer involves relating the source and target input-output spaces to a latent space

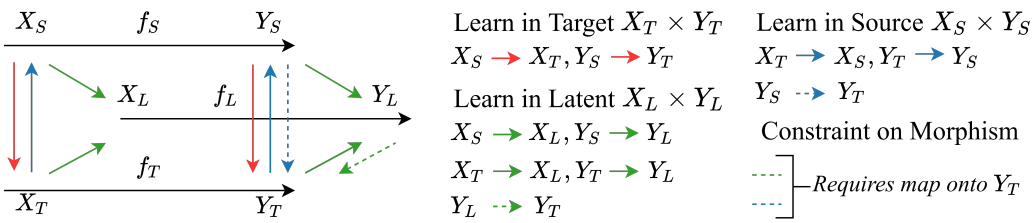


Figure 6: Morphisms in feature representation learning. Learning in the target sample space requires a morphism from that of the source, as shown in red. Learning in the source sample space requires a morphism from that of the target, as shown in blue, and a map from the source output to the target output, as shown by the dashed blue arrow. And learning in a latent sample space requires morphisms from both the source and target sample spaces to that of the latent system, as shown in green, and a map from the latent output to the target output, as shown by the dashed green arrow. As discussed in Section 5, the nature of these morphisms affects the difficulty of transfer.

$X_L \times Y_L$. Learning can occur in $X_L \times Y_L$, and, using m_{Y_L}, the output can be given in terms of the target output Y_T. Similarly, the target can be mapped onto the source, $X_L \times Y_L = X_S \times Y_S$, where learning can occur given m_{Y_L}, or the source can be mapped onto the target, $X_L \times Y_L = X_T \times Y_T$.

Figure 6 depicts these three cases of morphisms using a commutative diagram. As the individual maps that compose these morhpisms become more dislike identities and partial, feature-representation transfer becomes more difficult. We will discuss this further in our elaboration on structural considerations. Additionally note, even if $X_S \times Y_S = X_T \times Y_T$, feature-representation transfer may still be used to better relate source and target behavior.

Instance, parameter, and feature-based approaches are shown in terms of their specification on transfer learning algorithms A_{Tr} in Table 1. Another general notion in transfer learning approaches is *n*-shot transfer. It can be defined as follows.

Definition 12. N-shot Transfer.
A transfer learning system S_{Tr} with target data $d_T \in D_T$ is referred to as a n-shot transfer learning system if $|d_T| = n$. Zero-shot transfer occurs if $A_{Tr} : D \rightarrow \Theta_{Tr} \iff A_{Tr} : K_S \rightarrow \Theta_{Tr}$.

Machine learning is often concerned with few-shot learners—transfer learning systems that can generalize with only a few samples from the target. We will discuss generalization in transfer learning in the following subsection, but first, to get a sense of how we formalize instance, parameter, and feature-representation transfer, consider how a few canonical transfer learning algorithms are modeled by our

Transfer Approach	Algorithm Structure
Instance	$A_{Tr} : D_T \times D_S \rightarrow \Theta_{Tr}$
Parameter	$A_{Tr} : D_T \times \Theta_S \rightarrow \Theta_{Tr}$
Instance & Parameter	$A_{Tr} : D_T \times D_S \times \Theta_S \rightarrow \Theta_{Tr}$
Feature-Representation	$A_{Tr} : m_D(D) \rightarrow \Theta_{Tr}$

Table 1: Structural differences between transfer approaches.

framework.

Transfer component analysis uses a modified principal component analysis approach to project the source and target data into a relatable latent space [63], i.e., it is an instance approach in that D_S is used in A_{Tr} and a feature-representation approach in that X_S and X_T are projected into a latent X_L. Constraining parameters to be within a range of those of the source, as in hierarchical Bayesian and regularization approaches, is parameter transfer [26, 75]. Deep learning approaches often involve parameter transfer in that the weights Θ_S of the source network are shared and frozen in the target, or otherwise used to initialize Θ_T [7]. Other deep learning approaches also involve instance transfer to increase sample size, such as those that use generative adversarial networks [74]. When the source and target data must first be transformed before the data can be related, they are also feature-representation approaches, as in joint adaptation networks [47].

By formalizing the canonical classes of transfer approaches, we are better able to understand them in terms of their general requirements on S_{Tr}, particularly on S_I, and more particularly on A_{Tr} and D. The informal use of these classes by existing frameworks, wherein a solution method's dominant nature sorts it into a particular class, does well to organize the literature. Our formalisms can cloud these scholarly distinctions, as shown in the case of deep learning where a single method can belong to all three classes, however, they give a basis for defining formal categories of transfer learning systems S_{Tr} in terms of their inductive systems S_I.

3.3.3 Generalization in Transfer Learning

Generalization is, perhaps, the ultimate aim of learning. It is the ability for the learned hypothesis to approximate f out-of-sample, i.e., on samples not seen in training. Generalization as a goal for learning systems is implicit in A when a measure of error ϵ between $h(\theta)$ and f specifies G, such as in ERM. Herein, we

define it as follows.

Definition 13. Generalization.
Given a learning system S and data $d \in D$, generalization is the ability for a learned hypothesis $h(\theta)$ to estimate learning task $f : X \to Y$, on samples $(x, y) \notin d$.

In moving from the classical, identically distributed learning setting to transfer learning, we move from generalizing to a new sample from the same system, to generalizing to a new sample from a different system. In classical learning, for a learning system S, the estimated function f is specified by $P(Y|X)$ and data D are drawn from a related joint $P(X, Y)$. In transfer learning, however, the $X \times Y$ space and probability measures specifying f and D vary between S_S and S_T.

In classical learning, given a learning system S, data $d \in D$, a measure of error $\epsilon : H(\Theta) \times f \to \mathbb{R}$, and a threshold on error $\epsilon^* \in \mathbb{R}$, we generalize if

$$\epsilon(H(A(d)), f) \leq \epsilon^*.$$

That, is, if the measure of error between the learned hypothesis and the function it estimates is below a threshold. In practice, since f is not known, error is empirically estimated using samples $(x, y) \in X \times Y$ such that $(x, y) \notin d$.

In transfer learning, given S_{Tr} and data $d \in D$, we generalize if

$$\underbrace{\epsilon(H_{Tr}(A_{Tr}(d)), f_T)}_{\epsilon_T} \leq \epsilon^*.$$

If ϵ_T is smaller without any transferred knowledge from S_S than with, transfer from S_S to S_T is said to result in negative transfer. Negative transfer is defined in accord with Wang et. al as follows.

Definition 14. Negative Transfer.
Consider a transfer learning system S_{Tr}. Recall $D \subset D_T \times K_S$. Let $d \in D$ and $d_T \in D_T$. Given a measure of error $\epsilon : H(\Theta) \times f \to \mathbb{R}$, negative transfer is said to occur if

$$\epsilon(H_T(A_T(d_T)), f_T) < \epsilon(H_{Tr}(A_{Tr}(d)), f_T),$$

that is, if the error in estimating f_T is higher with the transferred knowledge than without it.

As Wang et. al note, negative transfer can arise from behavioral dissimilarity between the source and target [91]. In general, it can arise from structural dissimilarity as well.

Because generalization in transfer learning considers generalization across systems, as opposed to generalization within a given system, naturally, it is concerned

with the set of systems to and from which transfer learning can generalize. Using ϵ_T and ϵ^*, we can describe these sets as neighborhoods of systems *to* which we can transfer and generalize,

$$\underbrace{\{S_T | S_S, \epsilon_T \leq \epsilon^*\}}_{\text{Neighborhood of Targets } S_T}$$

and neighborhoods of systems *from* which we can transfer and generalize,

$$\underbrace{\{S_S | S_T, \epsilon_T \leq \epsilon^*\}}_{\text{Neighborhood of Sources } S_S} .$$

Noting Definition 14, if $\epsilon^* = \epsilon(H_T(A_T(d_T)), f_T)$, these neighborhoods are those systems to and from which transfer is positive.

The size of these neighborhoods describes the transferability of a learning system in terms of the number of systems it can transfer to or from and generalize. To the extent that cardinality gives a good description of size[2], transferability can be defined formally as follows.

Definition 15. Transferability.
Consider a target learning system S_T and a source learning system S_S. Given a measure of error $\epsilon_T : H_{Tr}(\Theta_{Tr}) \times f_T \to \mathbb{R}$ and a threshold on error $\epsilon^* \in \mathbb{R}$, the transferability of a source is the cardinality of the neighborhood of target systems S_T to which it can transfer and generalize,

$$|\{S_T | S_S, \epsilon_T \leq \epsilon^*\}|,$$

and the transferability of a target is the cardinality of the neighborhood of source systems S_S from which we can transfer and generalize,

$$|\{S_S | S_T, \epsilon_T \leq \epsilon^*\}|.$$

These cardinalities are termed the source-transferability and target-transferability, respectively.

Note, this defines transferability as an attribute of a particular system—not an attribute of a source-target pairing.

Our interest in transferability as an aim of transfer learning systems echoes a growing interest of the machine learning community in a notion of *generalist* learning systems [32, 39, 84]. Put informally, generalists are learning systems which can generalize to many tasks with few samples. Using our formalism, these systems can be described as learning systems with high source-transferability. More particularly, they can be defined as follows.

[2]Cardinality counts arbitrarily close systems as different, and it may be preferable to define a measure of equivalence, and consider the cardinality of the neighborhoods after the equivalence relation is applied.

Definition 16. Generalist Learning Systems.
A generalist learning system S_S is a system that can transfer to at least t target systems S_T with data $d_T \in D_T$ and generalize with at most n target samples $(x_T, y_T) \in X_T \times Y_T$. That is, they are systems S_S where

$$|\{S_T | S_S, |d_T| \leq n, \epsilon_T < \epsilon^*\}| \geq t$$

Otherwise put, generalists are sources S_S that can n-shot transfer learn to t or more targets S_T. Generalists are typically studied in the context of deep learning for computer vision, where a single network is tasked with few-shot learning a variety of visual tasks, e.g., classification, object detection, and segmentation, in a variety of environments [39].

In the following, we go beyond existing frameworks to explore notions of transferability—and thereby generalization, transfer roughness, and transfer distance in the context of structure and behavior. In doing so, we demonstrate the mathematical depth of Definition 8. We show that not only does it allow for immediate, formal consideration of surface-level phenomena covered by existing frameworks, but moreover, it allows for a considerable amount of modeling to be done at the general level, i.e., without reference to solution methods, in following with the spirit of AST depicted in Figure 1.

3.4 Structure and Behavior in Transfer Learning

To the extent that generalization in transfer learning is concerned with sets of systems, it is concerned with how those sets can be expressed in terms of those systems' structures and behaviors. In the following subsections, we discuss how structural and behavioral equality and, moreover, similarity relate to the difficulty of transfer learning. Equalities between S_S and S_T give a basic sense of the setting and what solution methods are available. Similarities between S_S and S_T are a richer means for elaboration, and can give a sense of the likelihood of generalization.

Learning systems are concerned with estimating functions $f : X \to Y$. As transfer learning is concerned with sharing knowledge used to estimate a source function $f_S : X_S \to Y_S$ to help estimate a target function $f_T : X_T \to Y_T$, naturally, the input-output spaces of the source $X_S \times Y_S$ and target $X_T \times Y_T$ are the principal interest of structural considerations. Similarly, the principal interest of behavioral considerations are the probability measures which specify f_S and f_T, and, correspondingly, D_S and D_T.

3.4.1 Structural Considerations

For source and target systems S_S and S_T we have the following possible equalities between system structures:

$$\begin{aligned} X_S = X_T, Y_S = Y_T, \\ X_S \neq X_T, Y_S = Y_T, \\ X_S = X_T, Y_S \neq Y_T, \\ X_S \neq X_T, Y_S \neq Y_T. \end{aligned}$$

The first case $X_S \times Y_S = X_T \times Y_T$ specifies transfer as homogeneous—all others specify heterogeneous transfer. This is the extent of discussion of structure in the existing frameworks [64, 93]. We elaborate further.

To do so, we extend past structural equality to notions of structural similarity. Recall, structural similarity is a question of the structural homomorphism between two systems. As is common in category theory, we define a morphism as simply a map between systems, and define an onto map between systems as a homomorphism. We can investigate homomorphism in reference to a morphism $m : S_S \to S_T$. First, note that we can quantify structural similarity using equivalence classes. Let $m_x : X_S \to X_T$ and $m_y : Y_S \to Y_T$ such that $m \leftrightarrow (m_x, m_y)$. And let S_S/m, X_S/m_x, and Y_S/m_y be the equivalence classes of S_S, X_S, and Y_S with respect to m, m_x, and m_y, respectively.

Consider the two sets of relations

$$\begin{aligned} w &: S_S \to S_S/m & z &: S_S/m \to S_T \\ w_x &: X_S \to X_S/m_x & z_x &: X_S/m \to X_T \\ w_y &: Y_S \to Y_S/m_y & z_y &: Y_S/m \to Y_T \end{aligned}$$

Relation w maps the source S_S to its equivalence class S_S/m and relation z maps S_S/m to the target S_T, as depicted by the commutative diagram shown in Figure 7. That is,

$$S_S \xrightarrow[(w_x, w_y)]{} S_S/m \xrightarrow[(z_x, z_y)]{} S_T$$

The equivalence class S_S/m describes the 'roughness' of the structural similarity from S_S to S_T. Its cardinality quantifies the 'surjective-ness' of $m : S_S \to S_T$. The greater the difference between $|S_S|$ and $|S_S/m|$, the more structurally dissimilar S_S and S_T are. However, in the large, structural similarity is not measurable in the same way as behavioral similarity.

The homomorphism between S_S and S_T is better investigated in terms of the properties of m, such as whether it is injective, surjective, invertible, etc. For example, partial morphisms from $X_S \times Y_S$ to $X_T \times Y_T$ are associated with partial

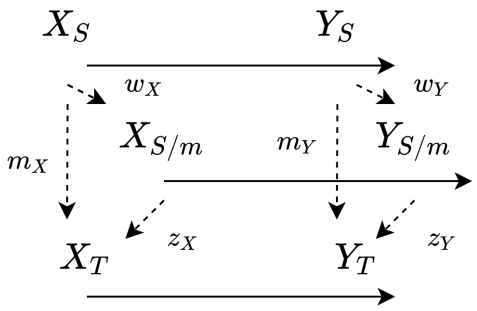


Figure 7: A commutative diagram depicting how equivalence classes can describe roughness.

transfer [13]. When the partial morphism is surjective, only a subset of the source is transferred to the target. When the partial morphism is injective, the source transfers to only a subset of the target. Also, structural similarity can be expressed using category theory, where the structural similarity between two systems can be studied with respect to the categories of systems to which they belong. To describe structural similarity in a broad sense, we define *transfer roughness* as follows.

Definition 17. Transfer Roughness.
Transfer roughness describes the structural homomorphism from the source system S_S to the target system S_T. When S_S and S_T are isomorphic, transfer roughness is minimal or otherwise non-existent. When roughness exists, it is defined by its properties, and thus there is no clear notion of maximal roughness.

The structure of the source relative to that of the target determines the roughness of transfer. Structures can be too dissimilar to transfer no matter what the behavior. Homomorphisms are onto and thus structure preserving, and, as such, it is a reasonable principle to characterize structural transferability in terms of the set of homomorphisms shared between the source and target. The supporting intuition is that either the source must map onto the target or they must both map onto some shared latent system, if not fully, at least in some aspect. Otherwise information in the source is lost when transferring to the target.

Let $\mathcal{H}(X, Y)$ denote the set of all structures homomorphic to $X \times Y$. The set of homomorphic structures between S_S and S_T is given by,

$$\mathcal{H}(X_S, Y_S) \cap \mathcal{H}(X_T, Y_T).$$

In transfer learning, we are specifically interested in using knowledge from S_S to help learn f_T. Thus, not all elements of this intersection are valid structures for

transfer learning, only those whose output can be mapped to Y_T. This set of valid structures can be expressed as,

$$\mathcal{V} = \{X \times Y \in \mathcal{H}(X_S, Y_S) \cap \mathcal{H}(X_T, Y_T) | \exists m_y : Y \to Y_T\}.$$

Apparently not all elements of $\mathcal{V}$ will be useful structures for estimating f_T, however, those that are useful, presuming structural homomorphism is necessary, will be in $\mathcal{V}$.

If we define $\mathcal{V}'$ to be the subset of $\mathcal{V}$ where transfer learning generalizes, i.e., the homomorphic structures where $\epsilon_T < \epsilon^*$, transferability can be defined in structural terms as follows.

Definition 18. Structural Transferability.
Consider a target learning system S_T and a source learning system S_S. The structural transferability of a source S_S is,

$$|\{S_T | S_S, \exists (X \times Y) \in \mathcal{V}'(S_S, S_T)\}|,$$

and the structural transferability of a target is,

$$|\{S_S | S_T, \exists (X \times Y) \in \mathcal{V}'(S_S, S_T)\}|.$$

In other words, structural transferability concerns the set of systems that share a useful homomorphism with S_S and S_T. While in practice $\mathcal{V}$ and $\mathcal{V}'$ are difficult to determine, they provide a theoretical basis for considering whether transfer learning is structurally possible between two systems and the structural invariance of the usefulness of transferred knowledge, respectively.

The relation $\mathcal{V}' \subset \mathcal{V}$ is particularly difficult. Ordering structural usefulness by homomorphism alone is difficult because of the vagueness of how homomorphism can be measured. The more isomorphism there is between S_S and S_T, the more the question of usefulness shifts to the behavior. There, the error ϵ provides the ordering[3] and the threshold ϵ^* provides the partition. Structural similarity provides no clear parallel.

It is true that if no homomorphism exists between S_S and S_T, they are from different categories. While functors can be used to map between categories, they necessarily distort transferred knowledge because they must add or remove structure to do so. Homomorphisms between systems, in contrast, are structure preserving. And so perhaps a partial order between homomorphic and non-homomorphic systems is justified. But this ordering is hardly granular. A more formal digression on this topic is beyond the scope of this paper, but well within the scope of AST [56].

[3] ϵ is a transfer distance between posteriors specifying $h(\theta)$ and f.

Example 3.3. *Transfer Roughness in UAVs.*
Consider S_S, S_T, and S_{Tr} defined according to Example 3.2. From Example 3.1 $X_S \times Y_S = X_T \times Y_T$, so S_{Tr} involves homogeneous transfer. But, if X_T did not include radar, transfer would be heterogeneous. Similarly so if Y_S described paths up to 100 meters in length and Y_T paths up to 10 meters. In either case, $X_S \times Y_S$ can map onto $X_T \times Y_T$, but $X_T \times Y_T$ cannot map onto $X_S \times Y_T$. Thus, transfer from S_T to S_S is rougher than transfer from S_S to S_T.

3.4.2 Behavioral Considerations

In transfer learning, the primary behaviors of interest are $P(X)$ and $P(Y|X)$ from the domain $\mathcal{D}$ and task $\mathcal{T}$, respectively, and the joint distribution they form,

$$P(X, Y) = P(X)P(Y|X).$$

It is important to realize that $P(X_S, Y_S) \neq P(X_T, Y_T)$ only implies that $P(X_S) \neq P(X_T) \vee P(Y_S|X_S) \neq P(Y_T|X_T)$. That is, the posteriors $P(Y|X)$ can still be equal when the joints $P(X, Y)$ are not if the marginals $P(X)$ offset the difference, and vice versa. In the main, these behavioral equalities make absolute statements on the inductive or transductive nature of a transfer learning system. Behavioral similarities, in contrast, have the richness to make statements on the likelihood of generalization, and, thereby, on transferability.

In AST, behavior is a topological-type concept and, accordingly, behavioral similarity is akin to a generalized metric. However, because in transfer learning we are concerned primarily with behaviors which are probability measures, behavioral similarity between S_S and S_T takes the form of distributional divergences. In our elaboration of behavioral similarity we focus on a notion of *transfer distance*. Transfer distance is the abstract distance knowledge must traverse to be transferred from one system to another. We consider it to be a measure on the input spaces $X_S \times X_T$, output spaces $Y_S \times Y_T$, or input-output spaces $(X_S \times Y_S) \times (X_T \times Y_T)$—more specifically, as a measure on probability measures over those spaces. It can be defined formally as follows.

Definition 19. Transfer Distance.
Let S_S and S_T be source and target learning systems. Let Z_i be a non-empty element of $\mathcal{P}(X_i \cup Y_i)$. Transfer distance δ_T is a measure

$$\delta_T : P(Z_S) \times P(Z_T) \to \mathbb{R}$$

of distance between the probability measures $P(Z_i)$ related to the estimated functions $f_i : X_i \to Y_i$ of S_S and S_T.

In practice, transfer distances are often given by f-divergences [22], such as KL-divergence or the Hellinger distance, Wasserstein distances [77], and maximum mean discrepancy [34, 47, 62]. Others use generative adversarial networks, a deep learning distribution modeling technique, to estimate divergence [30, 86]. Commonly, these distances are used to calculate divergence-based components of loss functions. Herein, we consider transfer distance's more general use in characterizing transfer learning systems.

In heterogeneous transfer, transfer distances can be used after feature-representation transfer has given the probability measures of interest the same support. Transfer distances between measures with different support are not widely considered in existing machine learning literature. However, the assumptions of homogeneous transfer and domain adaptation, i.e., $X_S \times Y_S = X_T \times Y_T$, allow for a rich theory of the role of transfer distance in determining the upper-bound on error.

Upper-bounds on ϵ_T have been given in terms of statistical divergence [10], H-divergence [5], Rademacher complexity [59], and integral probability metrics [102], among others. Despite their differences, central to most is a transfer distance $\delta_T : P(X_S) \times P(X_T) \to \mathbb{R}$ that concerns the closeness of input behavior and a term C that concerns the complexity of estimating f_T. These bounds roughly generalize to the form,

$$\epsilon_T \leq \epsilon_S + \delta_T + C$$

where ϵ_T and ϵ_S are the errors in S_T and S_S, δ_T is the transfer distance, and C is a constant term. C is often expressed in terms of sample sizes, e.g., $|D_S|$ and $|D_T|$, capacity, e.g., the VC-dimension of H_T [5], and information complexity, e.g., the Rademacher complexity of D_T [59]. Note, closeness and complexity are often not as separable as suggested by Inequality 1.

To the extent that Inequality 1 holds, we can describe transferability in terms of transfer distance. Generalization in transfer learning occurs if $\epsilon_T \leq \epsilon^*$, and since $\epsilon_T \leq \epsilon_S + \delta_T + C$, $\epsilon_S + \delta_T + C \leq \epsilon^* \implies \epsilon_T \leq \epsilon^*$. Thus, transferability can be defined in behavioral terms as follows.

Definition 20. Behavioral Transferability.
Consider a target learning system S_T and a source learning system S_S. The behavioral transferability of a source S_S is,

$$|\{S_T | S_S, \epsilon_S + \delta_T + C < \epsilon^*\}|,$$

and the behavioral transferability of a target is,

$$|\{S_S | S_T, \epsilon_S + \delta_T + C < \epsilon^*\}|.$$

For sources S_S with similar ϵ_S and targets S_T with similar C, given a threshold on distance $\delta^* \in \mathbb{R}$, behavioral transferability can be expressed entirely in terms of transfer distance:

$$|\{S_T|S_S, \delta_T < \delta^*\}| \text{ and } |\{S_S|S_T, \delta_T < \delta^*\}|.$$

Of course, specific bounds on ϵ_T with specific distances δ_T from the literature can be substituted in the stead of Inequality 1. Also note, we are assuming $X_S \times Y_S = X_T \times Y_T$. When $X_S \times Y_S \neq X_T \times Y_T$, transfer distance is a measure between probability measures with different supports, and while an upper-bound like Inequality 1 may be appropriate, it is not supported by existing literature. In such cases it is important to consider structural similarity.

Example 3.4. *Transfer Distance in UAVs.*
Consider S_S, S_T, and S_{Tr} defined according to Example 3.2. Let source S_S be associated with a desert biome and S_T a jungle biome. When comparing $P(X_T)$ to $P(X_S)$, increased foliage in S_T suggests accelerometer readings with higher variance, camera images with different hue, saturation, and luminance, and radar readings with more obstacles. Similarly, increased foliage may also mean paths in $P(Y_T|X_T)$ must compensate more for uncertainty than those in $P(Y_S|X_S)$. In contrast, foliage is more similar between the desert and tundra, thus, transfer distance is likely larger from the desert to the jungle than from the desert to the tundra.

3.4.3 Remarks

In summary, structure and behavior provide a means of elaborating deeply on transfer learning systems, just as they do for systems writ large. Structural considerations center on the structural relatability of S_S and S_T and the usefulness of the related structures $X \times Y$ for transfer learning. Behavioral considerations center on the behavioral closeness of S_S and S_T and the complexity of learning f_T. These concerns provide guideposts for the design and analysis of transfer learning systems. While the joint consideration of structure and behavior is necessary for a complete perspective on transfer learning systems, herein, in following with broader systems theory, we advocate that their joint consideration ought to come from viewing structure and behavior as parts of a whole—instead of approaching their joint consideration directly by neglecting notions of structure and behavior entirely, as is advocated implicitly by the existing frameworks pervasive use of domain $\mathcal{D}$ and task $\mathcal{T}$.

3.5 Conclusion

Our framework synthesizes systems theoretic notions of structure and behavior with key concepts in transfer learning. These include homogeneous and heterogeneous transfer, domain adaptation, inductive and transductive transfer, negative transfer, and more. In subsequent elaborations, we provide formal descriptions of transferability, transfer roughness, and transfer distance, all in reference to structure and behavior.

This systems perspective places emphasis on different aspects of transfer learning than existing frameworks. When we take behavior to be represented by a posterior or joint distribution, we arrive at constructs similar to existing theory. More distinctly, when we introduce structure, and study it in isolation, we arrive at notions of roughness, homomorphism, and category neglected in existing literature.

The presented framework offers a formal approach for modeling learning. The focal points of our theory are in aspects central to the general characterization and categorization of transfer learning as a mathematical construct, not aspects central to scholarship. This strengthens the literature by contributing a framework that is more closely rooted to engineering design and analysis than existing frameworks. Because our framework is pointedly anchored to concepts from existing surveys, practitioners should face little difficulty in the simultaneous use of both. Taken together, practitioners have a modeling framework and a reference guide to the literature.

Herein, we have modeled transfer learning as a subsystem. Transfer learning systems can be connected component-wise to the systems within which they are embedded. Subsequently, deductions can be made regarding the design and operation of systems and their learning subsystems with the interrelationships between them taken into account. In this way, we contribute a formal systems theory of transfer learning to the growing body of engineering-centric frameworks for machine learning. In the following, we explore the use of transfer distance in system design and operation.

4 Transfer Distance for System Design and Operation

4.1 Introduction

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

Applied machine learning for prognostics and health management (PHM) is prototypical of this trend and challenge. Non-stationarities 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 is 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. Additionally, 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 over-reliance on data [70].

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, we suggest a focus on methodologies that support the design and operation of systems to limit non-stationarities 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 non-stationarities 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 AI.

This section is structured as follows. First, we provide additional background on concept drift, PHM, and computer vision in Section 4.2. 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 in Section 4.3. Subsequently, we apply our methodology to characterize the transfer learning problems induced by an actuator rebuild procedure and mission deployment in Section 4.4 and 4.5. We conclude with a synopsis and a statement of future work in Section 4.6.

4.2 Background

We briefly review concept drift, PHM, and computer vision, and note this section's relationship to them. In short, this Section 4 presents PHM and computer vision case studies in empirically characterizing transfer distance using principles from domain adaptation and methods from concept drift.

4.2.1 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 a concept, to another. There are many metrics similar to transfer distance used in the concept drift literature to characterize drift [92]. Drift in these streaming systems has been modeled and simulated using Gaussian mixture models [21, 95]. Many methods use Hellinger distance to calculate distributional divergence because it is bounded $[0, 1]$ and symmetric [22, 92]. Consistent with concept drift literature, we use a combination of Gaussian mixture models and Hellinger distance to characterize distributional change.

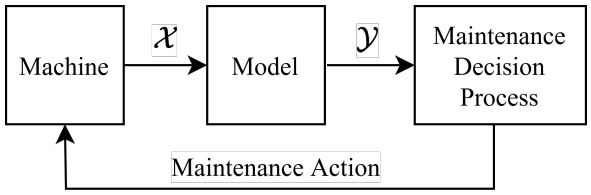


Figure 8: Data-driven models inform maintenance actions which change the distribution of their data. Non-stationarity is inherent in PHM.

4.2.2 Prognostics and Health Management

PHM is concerned with the use of prognostics and diagnostics for the management of machine health [85]. 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 [45], PHM is crucial to economic productivity. Furthermore, PHM helps safeguard critical systems such as gears in rotorcraft [20], whose failure can cause loss of propulsion mid-flight, and air filtration systems [41], whose failure in high pressure environments such as submarines can be equally catastrophic, among others [25].

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 [104]. 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 non-stationarities and label constraints [46, 48, 76, 100, 103].

Non-stationarity is a fundamental challenge in PHM. In data-driven PHM, sensor data from machines is 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 represented in Figure 8. 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 non-stationarity 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 [2, 3, 27, 51]. 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 [1]. 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 [17]. 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.

4.2.3 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 [60, 99]. Instead of extracting expert-defined features from images as a pre-processing 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 data sets [31].

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 [14]. 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 auto-encoder for dimension reduction, similar to existing approaches to explainable AI [12].

4.3 Methods

Transfer distance is usually referred to informally, e.g., to describe *near* or *far* transfer. It is implicit in the use of Wasserstein distance [77], maximum mean discrepancy [47, 62], generative adversarial networks [30, 86], and others, to calculate distributional-divergence-based components of loss functions in transfer learning

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

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.

Definition 19 of transfer distance directs our interest towards 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 $P(Y|X)$. Note, due to the focus on probability, we distinguish X and Y as random variables corresponding to sample spaces $\mathcal{X}$ and $\mathcal{Y}$, respectively, in contrast to Section 3.

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 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 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 Gaussian mixture models (GMMs) to fit the likelihoods $P(X|Y)$, i.e., as the *fitter* method in *fit* function of Algorithm 1. Gaussian mixture modeling is a clustering technique whereby a mixture of probability weighted multi-variate Gaussian distributions is fit to data. Each point is assigned to a single multi-variate 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 multi-variate Gaussian distribution with mean μ_k and co-variance σ_k.

Explicit, 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 Kullback-Leibler (KL) divergence as our transfer distances δ. Given two discrete probability distributions $P = (p_1, ..., p_n)$ and $Q = (q_1, ..., q_n)$, 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

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

KL is not symmetric and is unbounded above $[0, \infty)$, where its lower bound implies that the distributions are completely identical.

4.4 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,

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

suggests that a higher transfer distance δ is associated with a higher demand on the error in the source ϵ_S and the constant term C to keep the upper bound on error in the target ϵ_T the same as with a lower transfer distance. Therefore, transfer distance has a strong, fundamental influence on generalization difficulty.

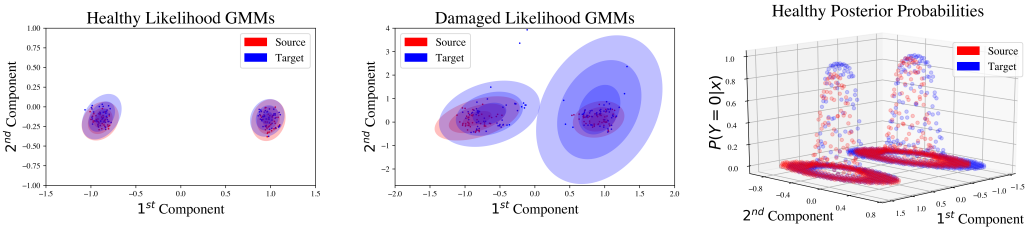
In cases where the distance between the source and target is, for example, associated with some physical change in the system, we can use transfer distance as a means of associating the physical change with generalization difficulty. Consider the generalization of prognostics models across system rebuilds. 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, when the actuator is deconstructed and rebuilt, the same classifier does marginally better than random guessing. When transfer learning is applied classification accuracy recovers to almost 90% [17].

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 multi-class 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 re-simulated. 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 pre-processing, to capture aspects of time-dependence, the data is 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.

4.4.1 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



(a) $p(X|Y=0)$ (b) $p(X|Y=1)$ (c) $P(Y=0|X)$

Figure 9: Likelihood Densities and Healthy Posterior Distributions

healthy and 1480 damage samples in the source, and 1098 and 1822 in the target, respectively.

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.

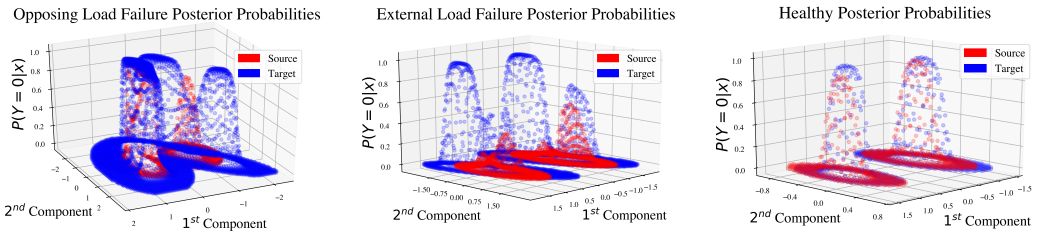
The fitted likelihoods and the constructed posterior probability of being healthy are plotted in Figure 9. The likelihood densities in Figures 9a and 9b show the source in red and target in blue, fit with 2-component Gaussian mixture models, where each concentric ellipse represents 1 standard deviation 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 Figure 9c 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 2. 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 the prior odds of failure decrease, δ_X and $\delta_{Y=0|X}$ decreases 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

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

Table 2: Hellinger transfer distance for relevant distributions. Note, $\delta_{X|Y}$ does not depend on $P(Y)$.



(a) $p(X|Y=1)$ (b) $p(X|Y=2)$ (c) $P(Y=5|X)$

Figure 10: Posterior Distributions for Different Failure Modes

samples. The transfer distance between the healthy source and target data suggests a much smaller change than 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.

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 also, it allows transfer learning algorithms to specifically focus on transferring knowledge for those failure modes which need source knowledge the most.

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

Table 3: Hellinger transfer distance for relevant distributions.

We quantify the transfer distance between failure modes in the source and target using a multi-class health state classification problem. Now, $\mathcal{Y} = \{0, 1, 2, 3, 4, 5\}$ where $Y = 0$ signifies healthy and $Y = 1, ..., 5$ 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 for $\forall y \in \mathcal{Y}$. Table 3 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.

Figure 10 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 3. 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 PCA space.

4.4.2 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

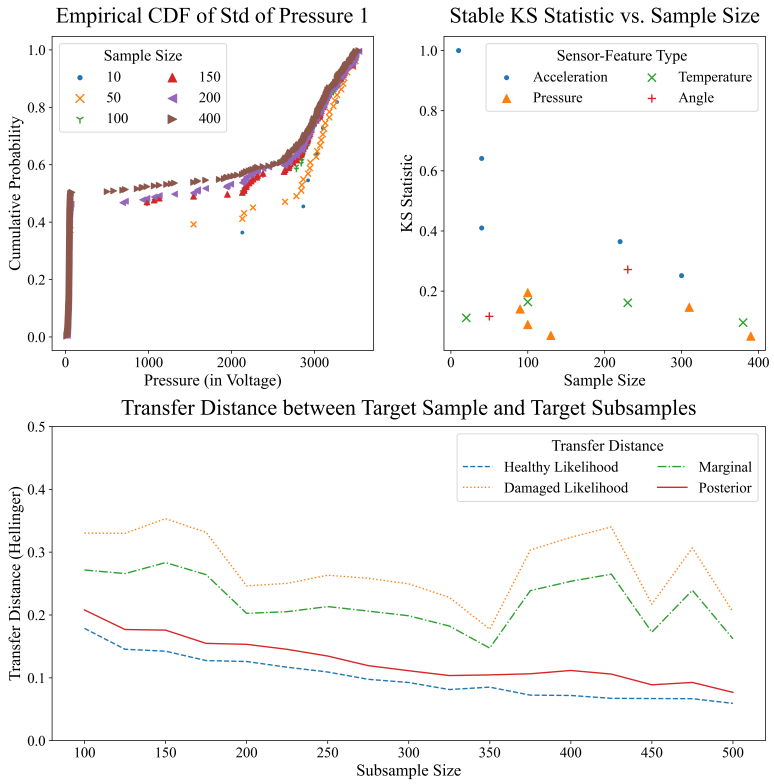


Figure 11: The top-left plot shows how the empirical cumulative distribution function (CDF) of the standard deviation of a pressure gauge changes with sample size. The top-right plot shows how many samples it takes for different sensor-feature types to converge to a stable value, labeled according to 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.

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 own sample complexity. Note the top-left plot in Figure 11 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 Figure 11 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 Figure 11 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 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.

4.5 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.

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 Southern and Northern California. In either case, 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

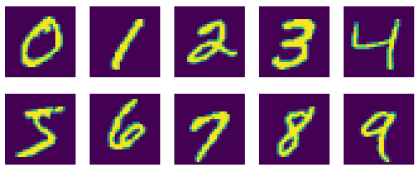


Figure 12: Example MNIST digit images 0-9 from left to right, top to bottom.

data set MNIST [42]. Then, with this understanding, we explore an application in defense where a model trained to detect the presence of aircraft in Southern California is deployed on a mission in Northern California [35]. In both cases we use auto-encoders to compress the images into a low-dimensional, latent representation before applying Algorithm 1 to compute transfer distances of interest. We use Gaussian mixture models as before, but now use KL divergence instead of Hellinger distance as our measure of transfer distance δ.

4.5.1 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 [4]. 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 data set contains examples of handwritten digits 0 thru 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 auto-encoder to the source images and use it to represent the source and target images as bivariate Gaussian distributions [37].

The transformed images are plotted in Figure 13 according to their Gaussian means μ. Whereas 0, 1, 6, and 7 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

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

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 Figure 14. While all show differences, both the source and target '0' and '6' images share some overlap. In contrast, the source and target '3' images 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 [61]. When we calculate the recall on the rotated, target images of a classifier trained on the non-rotated, 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 Figure 15. 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 transfer distance, whereas '8' and '1' images have the lowest recall and transfer distance. Recall decreases with transfer distance. 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.

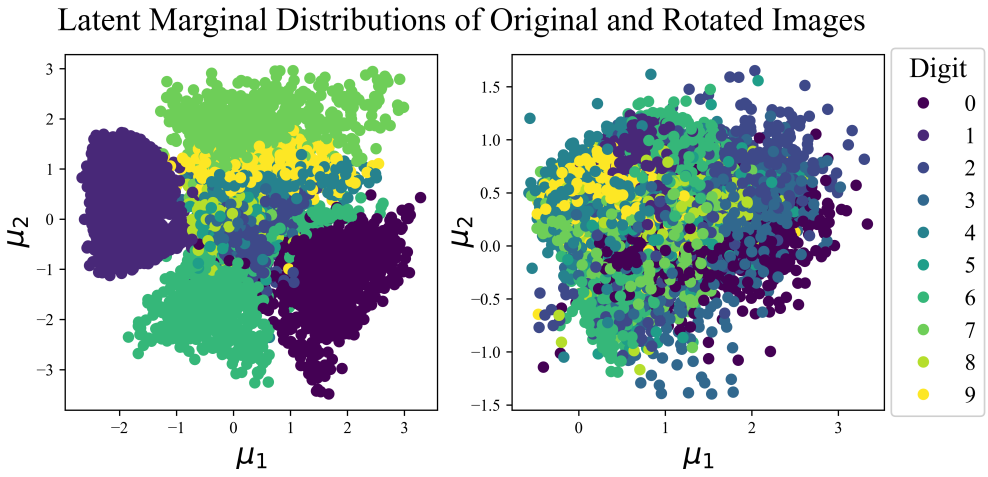


Figure 13: MNIST original, source images (left) and rotated, target images (right) in the variational auto-encoder's latent space.

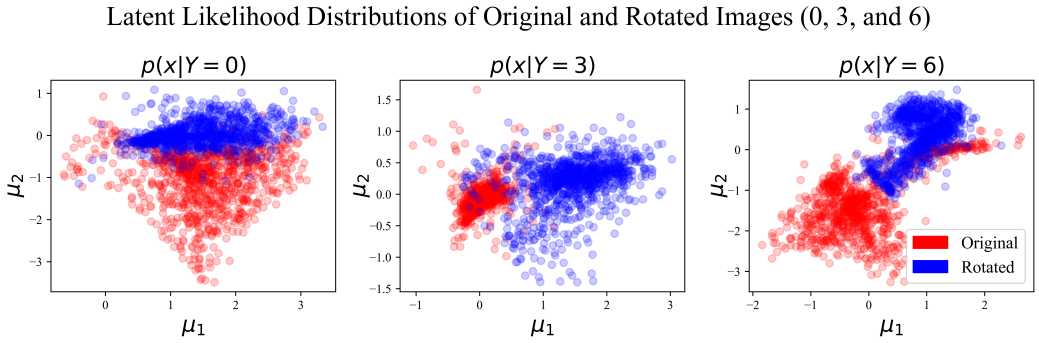


Figure 14: The variational auto-encoder's latent space shows the effect of rotation varies by digit.

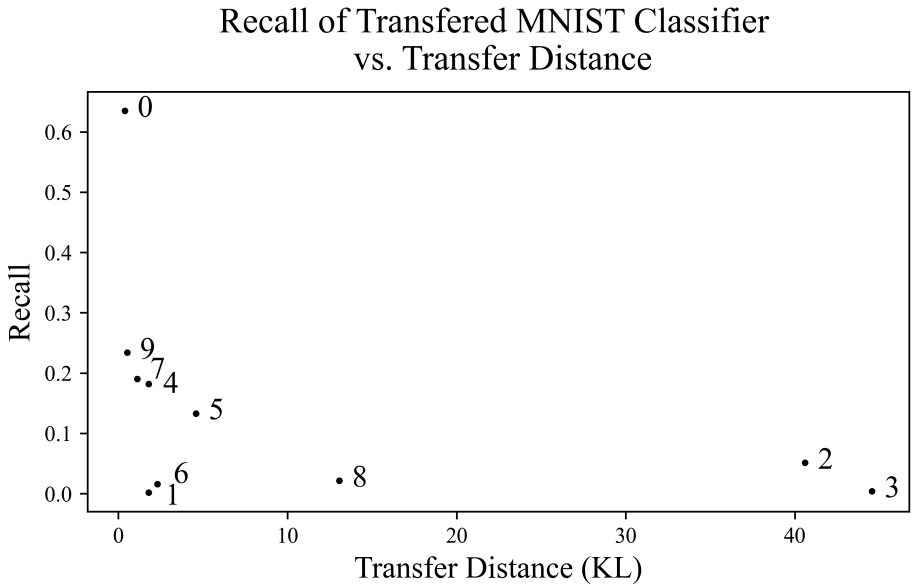


Figure 15: Higher recall digits tend to have lower transfer distance.

4.5.2 Mission Scenario in Aircraft Detection

Figure 16: Example aircraft images and non-aircraft images from California in the top and bottom rows, respectively.

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 Southern California and is tasked with operating in Northern California. Example images are shown in Figure 16. There are roughly 20000 images from Southern California and 12000 images from Northern California. We trained a convolutional neural network to detect aircraft on Southern California images.

When classifying held-out images from Southern California the classifier's accuracy is nearly 98%, but when classifying images from Northern California accuracy drops to nearly 85%, as shown in Figure 17. 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.

In order to apply our transfer distance methodology we first train an auto-encoder on the Southern California images. To do this, we initialize a convolutional auto-encoder with weights from the VGG-16 image classification network and then we fine-tune those weights [80]. We use the auto-encoder 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 data set, we batch the data into samples of 100 before fitting Gaussian mixture models.

When we calculate transfer distances δ_X between samples drawn from Southern

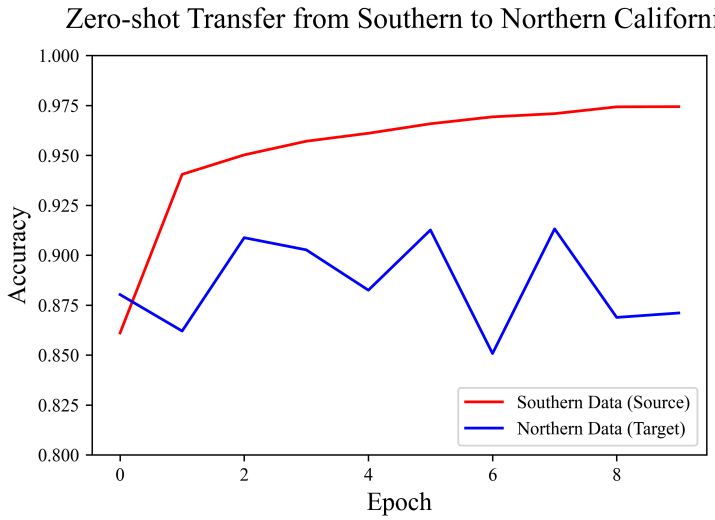
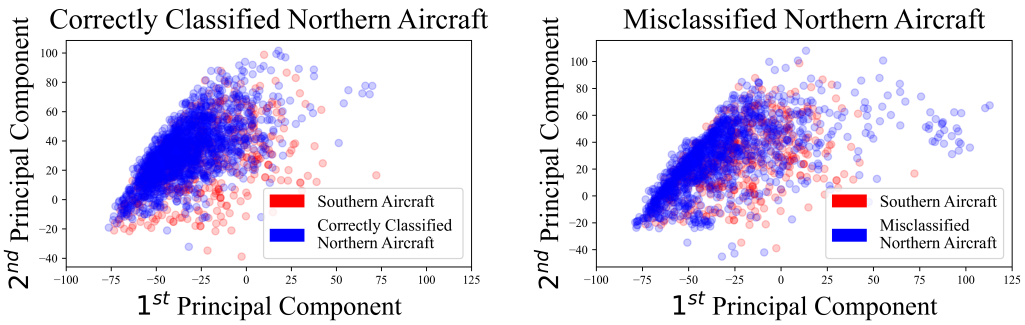


Figure 17: Shown is the classification accuracy of a convolutional neural network trained in Southern California evaluated on Southern California images, in blue, and on Northern California images, in orange, over the course of 10 training epochs.

California, we find them to have a mean KL divergence of 5.60. When we calculate transfer distances δ_X between samples drawn from Southern and Northern California, we find them to have a slightly higher mean KL divergence of 5.97. This suggests that samples drawn from Southern and Northern California are, on average, farther from each other than two samples drawn from Southern California. The small difference in expected transfer distance corresponds to the slight drop in classification accuracy in Figure 17.

We can investigate this trend by calculating transfer distances $\delta_{X|Y}$ of correctly and incorrectly classified images. 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 non-aircraft images from Northern California have a KL divergence of 2.34 from Southern California non-aircraft images, while misclassified non-aircraft images from 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 non-aircraft 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



(a) True positives

(b) False negatives

Figure 18: The first two principal components of true positives and false negative classifications when classifying images in Northern California using a classifier trained in Southern California. Misclassified Northern California aircraft images do not share a center of mass with Southern California aircraft images.

associated with the classifier trained on Southern California overhead imagery are shown in Figure 18a and 18b, 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 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 performance. Further, transfer distance between misclassified images is higher than that of 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.

4.6 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.

We demonstrated how to use transfer distance to compare the transferability of binary and multi-class health state classification. 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. Also, we showed how to determine the number of samples needed for stable estimates of transfer distance and transfer learning parameters, and suggested 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 which kinds 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 size data sets from different domains, i.e., sensor data and images, to highlight the generality of the methodology.

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.

5 Closing Remarks

Machine learning is heralded as a revolutionary technology. Its transition from a topic of research to a discipline of engineering is underway. Theoretical frameworks for characterizing learning systems in the context of the systems within which they are embedded are both understudied and essential to the principled application of machine learning technologies.

Herein we formulated a systems theory of transfer learning that is closely stitched to learning theory and machine learning, but takes a top-down, systems view. We showed that the resulting framework can be used to develop and organize best practices by showing how transfer distance can be used in system design and operation. We were able to integrate best practices from domain adaptation theory and quickly arrive at discipline-specific tradecraft for realizing best practice, such as the use of principal component analysis, Gaussian mixture models, and Hellinger distance in prognostics and health management and the use of auto-encoders and convolutional layers in computer vision. Therefore, the framework can be used both to make general considerations about best practices in learning systems and also to arrive at discipline-specific tradecraft for realizing those systems. And, thus, we showed that a systems theory of learning can serve as a foundation for a principled discipline of systems engineering for AI.

The limitations of the presented systems theoretic framework and systems engineering methodology stem from the level of abstraction with which they are concerned. The presented systems theory, in following with Mesarovician systems theory, is a minimal formalization of transfer learning. While the elaborations in Sections 3.3 and 3.4 add additional formalism, they do so at the general systems level of abstraction. Because, as demonstrated in Section 3.2, the presented systems theory is a super-structure for learning, these general systems findings apply to specific considerations of learning processes, the focus of learning theory, and learning algorithms, the focus of machine learning. But while research into new understandings of learning processes and algorithms can use the presented systems theory, that research is likely more parsimonious without the tedium of carrying along all of its set-theoretic formalism. If such research involves unconventional formulations, such as using topology or category theory instead of probability and linear algebra, the systems framework presented herein offers a foundation to build outwards from a shared base. That is, as long as the topology or category theory applies to learning systems as defined in Section 3.2, its findings should be consistent with existing probability theoretic and algebraic findings. Thus, the case for using the presented systems theory is strongest for meta-theoretical concerns and weakens the more specific a transfer learning phenomena of interest becomes.

Similar limitations apply to the presented systems engineering methodology.

Within the discipline of PHM or computer vision, there may be discipline-specific nuances which suggest exactly what distance measures to use. We highlighted the flexibility of our methodology to these nuances by applying it to both disciplines and using different distributional modeling techniques and different distance measures. This flexibility benefits the systems engineer by giving them a general measurement approach for any learning system, but necessitates a limitation on what can be specified about distributional modeling techniques, distance measures, etc., before the application area is known.

By building directly on the foundations of systems theory, the framework presented herein has natural, established ties to the practice of systems engineering. In future work, this connection to the broader body of systems research can be used to develop best practices for specification and certification, validation and testing, and life cycle engineering and sustainment of systems with learning algorithms. Furthermore, extending the application of the framework in system design and operation can help elucidate the role of systems engineers in machine learning engineering and the interplay between systems engineers, machine learning engineers, and others in achieving system and mission success in AI-heavy applications, as well as help develop the suite of tools available for facilitating such interdisciplinary work, e.g., transfer distance. Also, importantly, the systems theory of transfer learning presented herein suggests large gaps in the learning theory and machine learning literature involving notions of structure, homomorphism, and category. The set-theoretic structure of AST may provide a better means of closing this gap than the structure of the probability and optimization theory commonly found in learning theory and machine learning.

Real-world systems need transfer learning, and, correspondingly, engineering frameworks to guide its application. The presented framework offers a Mesarovician foundation.

6 Appendix

6.1 Mesarovician Glossary

Definition 21. System Behavior.
System behaviors are properties or descriptions paired with systems. For example, consider a system $S : X \to Y$ and a map $S \to \{stable, neutral, unstable\}$ or from $S \to P(X, Y)$. System behavior is a topological-type concept in the sense that it pairs systems with elements of sets of behaviors.

Definition 22. Behavioral Similarity.
Behavioral similarity describes the 'proximity' between two systems' behavior. To the extent that behavior can be described topologically, behavioral similarity can be expressed in terms of generalized metrics (topological 'distance'), metrics and pseudo-metrics (measure theoretic 'distance'), and statistical divergences (probability/information theoretic 'distance'), depending on the nature of the topology.

Definition 23. System Structure.
System structure is the mathematical structure of a system's component sets and the relations among them. For example, there may be algebraic structure, e.g. the linearity of a relationship between two component sets, related to the definition of the relation.

Definition 24. Structural Similarity.
Structural similarity describes the homomorphism between two systems' structures. It is described in reference to a relation $m : S_1 \to S_2$, termed a morphism. The equivalence class S_1/m describes the 'roughness' of the structural similarity between S_1 and S_2. Its cardinality gives a quantity to the 'surjective-ness' of $m : S_1 \to S_2$. However, in the large, structural similarity is not measurable in the same way as behavioral similarity. The homomorphism is better studied using properties of m.

Definition 25. Cascade Connection.
Let $\circ : \overline{S} \times \overline{S} \to \overline{S}$ be such that $S_1 \circ S_2 = S_3$, where,

$$S_1 \subset X_1 \times (Y_1 \times (Z_1)), S_2 \subset (X_2 \times Z_2) \times Y_2$$
$$S_3 \subset (X_1 \times X_2) \times (Y_1 \times Y_2), Z_1 = Z_2 = Z$$

and,

$$((x_1, x_2), (y_1, y_2)) \in S_3 \leftrightarrow$$
$$(\exists z)((x_1, (y_1, z)) \in S_1 \wedge ((x_2, z), y_2) \in S_2)$$

$\circ$ is termed the cascade (connecting) operator.

6.2 Learning Systems

Proposition. S in Definition 6 is a cascade connection of two input-output systems.
Proof: *Recall* $S \subset \times\{A, D, \Theta, H, X, Y\}$. *First we will show* A *and* H *to be input-output systems. First note that* $A \subset \times\{D, \Theta\}$. *Noting* $D \subset X \times Y$, *apparently* $D \cap \Theta = \emptyset$ *and* $D \cup \Theta = \overline{A}$. *Similarly,* $H \subset \times\{\Theta, X, Y\}$. *Letting* $X' = \{X, \Theta\}$, *apparently* $X' \cap Y = \emptyset$ *and* $X' \cup Y = \overline{H}$. *Therefore, by definition,* A *and* H *are input-output systems. Let* $S_C : D \times X \to Y$. *Apparently, for* $d \in D, x \in X, y \in Y, \theta \in \Theta$, $((d, x), y) \in S_C \leftrightarrow \exists\theta((d, \theta) \in A \wedge (\theta, x, y) \in H$. *Therefore,* $S_C : A \circ H$. *Lastly, note* S_C *is a function-type representation of* S, *where* A, H, *and* Θ *are left as specifications on relations, not included as component sets.*

Proposition. S in Definition 7 is a goal-seeking system.
Proof: *Goal-seeking is characterized by the consistency relations* (G, E) *and by the internal feedback of* $X \times Y$ *into* S_G. *Note* $D \subset X \times Y$ *satisfies internal feedback. The consistency relations* (G, E) *in Definition 3 and 7 can be shown to be isomorphic by substituting* $D \subset X \times Y$ *into consistency relations* G *and* E *in Definition 3 and* $(x, y) \in d$ *into their constraints. Thus, by definition,* S *in Definition 7 is a goal-seeking system, where* S_G *is the inductive system* A *and* S_F *is the functional system* H.

Proposition. Empirical risk minimization is a special case of a learning system as defined in Definition 7.
Proof: *A learning system given by Definition 7 is an empirical risk minimization learning system if (1)* D *is a sample of* l *independent and identically distributed observations sampled according to an unknown distribution* $P(X, Y)$, *and (2)* A *selects* $\theta \in \Theta$ *by minimizing the empirical risk* R_{emp}, *calculated on the basis of* D, *over* $\theta \in \Theta$. *Otherwise put, ERM is a learning system* $S \subset \times\{A, D, \Theta, G, E, H, X, Y\}$ *where* $G(D, \theta) = R_{emp}(D, \theta) = \frac{1}{l}\sum_{i=1}^{l} L(y_i, h(x_i, \theta))$ *and* $E = \min_{\theta\in\Theta} G(D, \theta)$, *where* L *is a loss function.*

References

[1] Stephen Adams, Peter A Beling, Kevin Farinholt, Nathan Brown, Sherwood Polter, and Qing Dong. Condition based monitoring for a hydraulic actuator. In *Annual Conference of the Prognostics and Health Management Society October 2016*, 2016.

[2] Stephen Adams, Ryan Meekins, Peter A Beling, Kevin Farinholt, Nathan Brown, Sherwood Polter, and Qing Dong. A comparison of feature selection and feature extraction techniques for condition monitoring of a hydraulic actuator. In *Annual Conference of the Prognostics and Health Management Society 2017*, 2017.

[3] Stephen Adams, Ryan Meekins, Peter A Beling, Kevin Farinholt, Nathan Brown, Sherwood Polter, and Qing Dong. Hierarchical fault classification for resource constrained systems. *Mechanical Systems and Signal Processing*, 134:106266, 2019.

[4] W Ross Ashby. *An introduction to cybernetics*. Chapman & Hall Ltd, 1961.

[5] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[6] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

[7] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 17–36, 2012.

[8] Ludwig von Bertalanffy. An outline of general system theory. *British Journal for the Philosophy of science*, 1950.

[9] Ludwig von Bertalanffy. The theory of open systems in physics and biology. *Science*, 111(2872):23–29, 1950.

[10] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In *Advances in neural information processing systems*, pages 129–136, 2008.

[11] Kenneth E Boulding. General systems theory—the skeleton of science. *Management science*, 2(3):197–208, 1956.

[12] S Bulusu, B Kailkhura, B Li, P Varshney, and D Song. Anomalous instance detection in deep learning: A survey. Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2020.

[13] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–150, 2018.

[14] Zhong Chen, Ting Zhang, and Chao Ouyang. End-to-end airplane detection using transfer learning in remote sensing images. *Remote Sensing*, 10(1):139, 2018.

[15] Tyler Cody, Stephen Adams, and Peter Beling. Motivating a systems theory of ai. *INSIGHT*, 23(1):37–40, 2020.

[16] Tyler Cody, Stephen Adams, and Peter A Beling. A systems theoretic perspective on transfer learning. In *2019 IEEE International Systems Conference (SysCon)*, pages 1–7. IEEE, 2019.

[17] Tyler Cody, Stephen Adams, Peter A Beling, Sherwood Polter, Kevin Farinholt, Nathan Hipwell, Ali Chaudhry, Kennet Castillo, and Ryan Meekins. Transferring random samples in actuator systems for binary damage detection. In *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 1–7. IEEE, 2019.

[18] Gabriela Csurka. A comprehensive survey on domain adaptation for visual applications. In *Domain adaptation in computer vision applications*, pages 1–35. Springer, 2017.

[19] Oscar Day and Taghi M Khoshgoftaar. A survey on heterogeneous transfer learning. *Journal of Big Data*, 4(1):29, 2017.

[20] Irebert R Delgado, Paula J Dempsey, and Donald L Simon. A survey of current rotorcraft propulsion health monitoring technologies. 2012.

[21] Javier Diaz-Rozo, Concha Bielza, and Pedro Larrañaga. Clustering of data streams with dynamic gaussian mixture models: an iot application in industrial processes. *IEEE Internet of Things Journal*, 5(5):3533–3547, 2018.

[22] Gregory Ditzler and Robi Polikar. Hellinger distance based drift detection for nonstationary environments. In *2011 IEEE symposium on computational intelligence in dynamic and uncertain environments (CIDUE)*, pages 41–48. IEEE, 2011.

[23] Gregory Ditzler, Manuel Roveri, Cesare Alippi, and Robi Polikar. Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, 10(4):12–25, 2015.

[24] Dov Dori, Hillary Sillitto, Regina M Griego, Dorothy McKinney, Eileen P Arnold, Patrick Godfrey, James Martin, Scott Jackson, and Daniel Krob. System definition, system worldviews, and systemness characteristics. *IEEE Systems Journal*, 2019.

[25] Maryam Eftekhari, Mehdi Moallem, Saeed Sadri, and Min-Fu Hsieh. Online detection of induction motor's stator winding short-circuit faults. *IEEE Systems Journal*, 8(4):1272–1282, 2013.

[26] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117, 2004.

[27] Kevin M Farinholt, Ali Chaudhry, Mark Kim, Ethan Thompson, Nathan Hipwell, Ryan Meekins, Stephen Adams, Peter Beling, and Sherwood Polter. Developing health management strategies using power constrained hardware. In *PHM Society Conference*, volume 10, 2018.

[28] Jay Wright Forrester. *Industrial dynamics*. MIT Press (Cambridge, Mass.), 1961.

[29] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):44, 2014.

[30] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[31] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[32] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *Advances in neural information processing systems*, pages 103–112, 2019.

[33] Jing Jiang. A literature survey on domain adaptation of statistical classifiers. *URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey*, 3:1–12, 2008.

[34] Min Jiang, Wenzhen Huang, Zhongqiang Huang, and Gary G Yen. Integration of global and local metrics for domain adaptation learning via dimensionality reduction. *IEEE transactions on cybernetics*, 47(1):38–51, 2015.

[35] Patcharin Kamsing, Peerapong Torteeka, and Soemsak Yooyen. Deep convolutional neural networks for plane identification on satellite imagery by exploiting transfer learning with a different optimizer. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 9788–9791. IEEE, 2019.

[36] Imen Khamassi, Moamar Sayed-Mouchaweh, Moez Hammami, and Khaled Ghédira. Discussion and review on evolving data streams and concept drift adapting. *Evolving systems*, 9(1):1–23, 2018.

[37] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[38] George J Klir. Approach to general systems theory. 1969.

[39] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 2019.

[40] Ilja Kuzborskij and Francesco Orabona. Stability and hypothesis transfer learning. In *International Conference on Machine Learning*, pages 942–950, 2013.

[41] Fakhreddine Landolsi, Hassene Jammoussi, and Imad Makki. Air filter diagnostics & prognostics in naturally aspired engines. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 61–65. IEEE, 2017.

[42] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[43] Christiane Lemke, Marcin Budka, and Bogdan Gabrys. Metalearning: a survey of trends and technologies. *Artificial intelligence review*, 44(1):117–130, 2015.

[44] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[45] Jingshan Li and Semyon M Meerkov. *Production systems engineering*. Springer Science & Business Media, 2008.

[46] Xudong Li, Yang Hu, Mingtao Li, and Jianhua Zheng. Fault diagnostics between different type of components: A transfer learning approach. *Applied Soft Computing*, 86:105950, 2020.

[47] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017.

[48] Weining Lu, Bin Liang, Yu Cheng, Deshan Meng, Jun Yang, and Tao Zhang. Deep model based domain adaptation for fault diagnosis. *IEEE Transactions on Industrial Electronics*, 64(3):2296–2305, 2017.

[49] Donald Macko. Natural states and past-determinism of general time systems. *Information Sciences*, 3(1):1–16, 1971.

[50] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[51] Ryan Meekins, Stephen Adams, Peter A Beling, Kevin Farinholt, Nathan Hipwell, Ali Chaudhry, Sherwood Polter, and Qing Dong. Cost-sensitive classifier selection when there is additional cost information. In *International Workshop on Cost-Sensitive Learning*, pages 17–30, 2018.

[52] D Mesarovic, Mihajlo, SN Sreenath, and JD Keene. Search for organising principles: understanding in systems biology. *Systems biology*, 1(1):19–27, 2004.

[53] Mihajlo D Mesarovic. Foundations for a general systems theory. *Views on general systems theory*, pages 1–24, 1964.

[54] Mihajlo D Mesarović. Systems theory and biology—view of a theoretician. In *Systems theory and biology*, pages 59–87. Springer, 1968.

[55] Mihajlo D Mesarovic and Yasuhiko Takahara. *General systems theory: mathematical foundations*, volume 113. Academic press, 1975.

[56] Mihajlo D Mesarovic and Yasuhiko Takahara. Abstract systems theory. 1989.

[57] Mihajlo D Mesarovic, Yasuhiko Takahara, and D Macko. Theory of hierarchical, multilevel systems. 1970.

[58] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[59] Mehryar Mohri and Afshin Rostamizadeh. Rademacher complexity bounds for non-iid processes. In *Advances in Neural Information Processing Systems*, pages 1097–1104, 2009.

[60] Loris Nanni, Stefano Ghidoni, and Sheryl Brahnam. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognition*, 71:158–172, 2017.

[61] Mahesh Pal. Random forest classifier for remote sensing classification. *International journal of remote sensing*, 26(1):217–222, 2005.

[62] Sinno Jialin Pan, James T Kwok, Qiang Yang, et al. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, pages 677–682, 2008.

[63] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2010.

[64] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[65] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[66] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015.

[67] Karl Popper. *The logic of scientific discovery*. Routledge, 2005.

[68] Anatol Rapoport. *General system theory: Essential concepts & applications*, volume 10. CRC Press, 1986.

[69] Anatol Rapoport and Samuel Ichiyé Hayakawa. Science and the goals of man: A study in semantic orientation. 1952.

[70] Charles Rogers, Jonathan Bugg, Chris Nyheim, Will Gebhardt, Brian Andris, Evan Heitman, and Cody Fleming. Adversarial artificial intelligence for overhead imagery classification models. In *2019 Systems and Information Engineering Design Symposium (SIEDS)*, pages 1–6. IEEE, 2019.

[71] Arturo Rosenblueth, Norbert Wiener, and Julian Bigelow. Behavior, purpose and teleology. *Philosophy of science*, 10(1):18–24, 1943.

[72] Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning*, volume 898, pages 1–4, 2005.

[73] Syed Moshfeq Salaken, Abbas Khosravi, Thanh Nguyen, and Saeid Nahavandi. Extreme learning machine based transfer learning algorithms: A survey. *Neurocomputing*, 267:516–524, 2017.

[74] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2018.

[75] Anton Schwaighofer, Volker Tresp, and Kai Yu. Learning gaussian process kernels via hierarchical bayes. In *Advances in neural information processing systems*, pages 1209–1216, 2005.

[76] Fei Shen, Chao Chen, Ruqiang Yan, and Robert X Gao. Bearing fault diagnosis based on svd feature extraction and transfer learning classification. In *Prognostics and System Health Management Conference (PHM), 2015*, pages 1–6. IEEE, 2015.

[77] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. *arXiv preprint arXiv:1707.01217*, 2017.

[78] Herbert A Simon. A study of decision-making processes in administrative organizations. *Administrative Behavior*, 1957.

[79] Herbert A Simon. *The sciences of the artificial*. 1969.

[80] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[81] Kate A Smith-Miles. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys (CSUR)*, 41(1):6, 2009.

[82] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[83] Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685, 2009.

[84] Michael Tschannen, Josip Djolonga, Marvin Ritter, Aravindh Mahendran, Neil Houlsby, Sylvain Gelly, and Mario Lucic. Self-supervised learning of video-induced visual invariances. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13806–13815, 2020.

[85] Kwok L Tsui, Nan Chen, Qiang Zhou, Yizhen Hai, and Wenbin Wang. Prognostics and health management: A review on data driven approaches. *Mathematical Problems in Engineering*, 2015, 2015.

[86] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015.

[87] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1995.

[88] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95, 2002.

[89] Ricardo Vilalta, Christophe Giraud-Carrier, and Pavel Brazdil. Meta-learning-concepts and techniques. In *Data mining and knowledge discovery handbook*, pages 717–731. Springer, 2009.

[90] Ludwig Von Bertalanffy. General system theory: Foundations, development, applications. 1969.

[91] Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11293–11302, 2019.

[92] Geoffrey I Webb, Roy Hyde, Hong Cao, Hai Long Nguyen, and Francois Petitjean. Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30(4):964–994, 2016.

[93] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):9, 2016.

[94] Norbert Wiener. *Cybernetics or Control and Communication in the Animal and the Machine*. MIT press, 2019.

[95] Jun Wu, Xian-Sheng Hua, and Bo Zhang. Tracking concept drifting with gaussian mixture model. In *Visual Communications and Image Processing 2005*, volume 5960, page 59604L. International Society for Optics and Photonics, 2005.

[96] A Wayne Wymore. *Systems engineering methodology for interdisciplinary teams*. John Wiley & Sons, 1976.

[97] A Wayne Wymore. *Model-based systems engineering*. CRC press, 1993.

[98] Wayne Wymore. A mathematical theory of systems engineering: the elements. 1967.

[99] Xiao Xiao, Dan Xu, and Wanggen Wan. Overview: Video recognition from handcrafted method to deep learning method. In *2016 International Conference on Audio, Language and Image Processing (ICALIP)*, pages 646–651. IEEE, 2016.

[100] Junyao Xie, Laibin Zhang, Lixiang Duan, and Jinjiang Wang. On cross-domain feature fusion in gearbox fault diagnosis under various operating conditions based on transfer component analysis. In *Prognostics and Health Management (ICPHM), 2016 IEEE International Conference on*, pages 1–6. IEEE, 2016.

[101] Qian Xu and Qiang Yang. A survey of transfer and multitask learning in bioinformatics. *Journal of Computing Science and Engineering*, 5(3):257–268, 2011.

[102] Chao Zhang, Lei Zhang, and Jieping Ye. Generalization bounds for domain adaptation. In *Advances in neural information processing systems*, pages 3320–3328, 2012.

[103] Wei Zhang, Gaoliang Peng, Chuanhao Li, Yuanhang Chen, and Zhujun Zhang. A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors*, 17(2):425, 2017.

[104] Weiting Zhang, Dong Yang, and Hongchao Wang. Data-driven methods for predictive maintenance of industrial equipment: a survey. *IEEE Systems Journal*, 13(3):2213–2227, 2019.

[105] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

[106] Indrė Žliobaitė. Learning under concept drift: an overview. *arXiv preprint arXiv:1010.4784*, 2010.

[107] Indrė Žliobaitė, Mykola Pechenizkiy, and Joao Gama. An overview of concept drift applications. In *Big data analysis: new algorithms for a new society*, pages 91–114. Springer, 2016.