Taskonomy: Disentangling Task Transfer Learning

Amir R. Zamir[1,2] Alexander Sax[1*] William Shen[1*] Leonidas Guibas[1] Jitendra Malik[2] Silvio Savarese[1]

[1] Stanford University [2] University of California, Berkeley

http://taskonomy.vision/

Abstract

*Do visual tasks have a relationship, or are they unrelated? For instance, could having surface normals simplify estimating the depth of an image? Intuition answers these questions positively, implying existence of a **structure** among visual tasks. Knowing this structure has notable values; it is the concept underlying transfer learning and provides a principled way for identifying redundancies across tasks, e.g., to seamlessly reuse supervision among related tasks or solve many tasks in one system without piling up the complexity.*

We proposes a fully computational approach for modeling the structure of space of visual tasks. This is done via finding (first and higher-order) transfer learning dependencies across a dictionary of twenty six 2D, 2.5D, 3D, and semantic tasks in a latent space. The product is a computational taxonomic map for task transfer learning. We study the consequences of this structure, e.g. nontrivial emerged relationships, and exploit them to reduce the demand for labeled data. We provide a set of tools for computing and probing this taxonomical structure including a solver users can employ to find supervision policies for their use cases.

1. Introduction

Object recognition, depth estimation, edge detection, pose estimation, etc are examples of common vision tasks deemed useful and tackled by the research community. Some of them have rather clear relationships: we understand that surface normals and depth are related (one is a derivate of the other), or vanishing points in a room are useful for orientation. Other relationships are less clear: how keypoint detection and the shading in a room can, together, perform pose estimation.

The field of computer vision has indeed gone far without explicitly using these relationships. We have made remarkable progress by developing advanced learning machinery (e.g. ConvNets) capable of finding complex mappings from

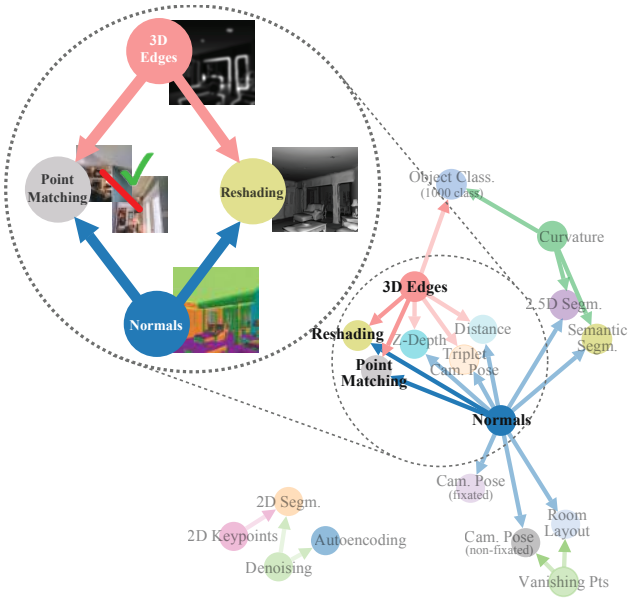


Figure 1: **A sample task structure discovered by the computational task taxonomy (*taskonomy*)**. It found that, for instance, by combining the learned features of a surface normal estimator and occlusion edge detector, good networks for reshading and point matching can be rapidly trained with little labeled data.

X to Y when many pairs of (x, y) s.t. $x \in X, y \in Y$ are given as training data. This is usually referred to as fully supervised learning and often leads to problems being solved in isolation. Siloing tasks makes training a new task or a comprehensive perception system a Sisyphean challenge, whereby each task needs to be learned individually from scratch. Doing so ignores their quantifiably useful relationships leading to a massive labeled data requirement.

Alternatively, a model aware of the relationships among tasks demands less supervision, uses less computation, and behaves in more predictable ways. Incorporating such a structure is the first stepping stone towards developing provably efficient comprehensive/universal perception models [32, 4], i.e. ones that can solve a large set of tasks before becoming intractable in supervision or computation demands. However, this task space structure and its effects

*Equal.

are still largely unknown. The relationships are non-trivial, and finding them is complicated by the fact that we have imperfect learning models and optimizers. In this paper, we attempt to shed light on this underlying structure and present a framework for mapping the space of visual tasks. Here what we mean by "structure" is a collection of computationally found relations specifying which tasks supply useful information to another, and by how much (see Fig. 1).

We employ a fully computational approach for this purpose, with neural networks as the adopted computational function class. In a feedforward network, each layer successively forms more abstract representations of the input containing the information needed for mapping the input to the output. These representations, however, can transmit statistics useful for solving other outputs (tasks), presumably if the tasks are related in some form [80, 17, 56, 44]. This is the basis of our approach: we computes an affinity matrix among tasks based on whether the solution for one task can be sufficiently easily read out of the representation trained for another task. Such transfers are exhaustively sampled, and a Binary Integer Programming formulation extracts a globally efficient transfer policy from them. We show this model leads to solving tasks with far less data than learning them independently and the resulting structure holds on common datasets (ImageNet [75] and Places [101]).

Being fully computational and representation-based, the proposed approach avoids imposing prior (possibly incorrect) assumptions on the task space. This is crucial because the priors about task relations are often derived from either human intuition or analytical knowledge, while neural networks need not operate on the same principles [60, 31, 38, 43, 99, 85]. For instance, although we might expect depth to transfer to surface normals better (derivatives are easy), the opposite is found to be the better direction in a computational framework (i.e. suited neural networks better).

An interactive taxonomy solver which uses our model to suggest data-efficient curricula, a live demo, dataset, and code are available at http://taskonomy.vision/.

2. Related Work

Assertions of existence of a structure among tasks date back to the early years of modern computer science, e.g. with Turing arguing for using learning elements [92, 95] rather than the final outcome or Jean Piaget's works on developmental stages using previously learned stages as sources [71, 37, 36], and have extended to recent works [73, 70, 48, 16, 94, 58, 9, 63]. Here we make an attempt to actually find this structure. We acknowledge that this is related to a breadth of topics, e.g. compositional modeling [33, 8, 11, 21, 53, 89, 87], homomorphic cryptography [40], lifelong learning [90, 13, 82, 81], functional maps [68], certain aspects of Bayesian inference and Dirichlet processes [52, 88, 87, 86, 35, 37], few-shot learning [78, 23, 22, 67, 83], transfer learning [72, 81, 27, 61, 64, 57], un/semi/self-supervised learning [20, 6, 15, 100, 17, 80], which are studied across various fields [70, 91, 10]. We review the topics most pertinent to vision within the constraints of space:

Self-supervised learning methods leverage the inherent relationships between tasks to learn a desired expensive one (e.g. object detection) via a cheap surrogate (e.g. colorization) [65, 69, 15, 100, 97, 66]. Specifically, they use a manually-entered local part of the structure in the task space (as the surrogate task is manually defined). In contrast, our approach models this large space of tasks in a computational manner and can discover obscure relationships.

Unsupervised learning is concerned with the redundancies in the input domain and leveraging them for forming compact representations, which are usually agnostic to the downstream task [6, 47, 18, 7, 30, 74]. Our approach is not unsupervised by definition as it is not agnostic to the tasks. Instead, it models the space tasks belong to and in a way utilizes the *functional* redundancies among tasks.

Meta-learning generally seeks performing the learning at a level higher than where conventional learning occurs, e.g. as employed in reinforcement learning [19, 29, 26], optimization [2, 79, 46], or certain architectural mechanisms [25, 28, 84, 62]. The motivation behind meta learning has similarities to ours and our outcome can be seen as a computational meta-structure of the space of tasks.

Multi-task learning targets developing systems that can provide multiple outputs for an input in one run [48, 16]. Multi-task learning has experienced recent progress and the reported advantages are another support for existence of a useful structure among tasks [90, 97, 48, 73, 70, 48, 16, 94, 58, 9, 63]. Unlike multi-task learning, we explicitly model the relations among tasks and extract a meta-structure. The large number of tasks we consider also makes developing one multi-task network for all infeasible.

Domain adaption seeks to render a function that is developed on a certain domain applicable to another [42, 96, 5, 77, 50, 24, 34]. It often addresses a shift in the *input* domain, e.g. webcam images to D-SLR [45], while the task is kept the same. In contrast, our framework is concerned with *output* (task) space, hence can be viewed as *task/output adaptation*. We also perform the adaptation in a larger space among many elements, rather than two or a few.

3. Method

We define the problem as follows: we want to maximize the collective performance on a set of tasks $\mathcal{T} = \{t_1, ..., t_n\}$, subject to the constraint that we have a limited supervision budget γ (due to financial, computational, or time constraints). We define our supervision budget γ to be the maximum allowable number of tasks that we are willing to train from scratch (i.e. *source* tasks). The task dictionary is defined as $\mathcal{V}$=$\mathcal{T} \cup \mathcal{S}$ where $\mathcal{T}$ is the set of tasks which we want solved (*target*), and $\mathcal{S}$ is the set of tasks that can be trained (*source*). Therefore, $\mathcal{T} - \mathcal{T} \cap \mathcal{S}$ are the tasks that

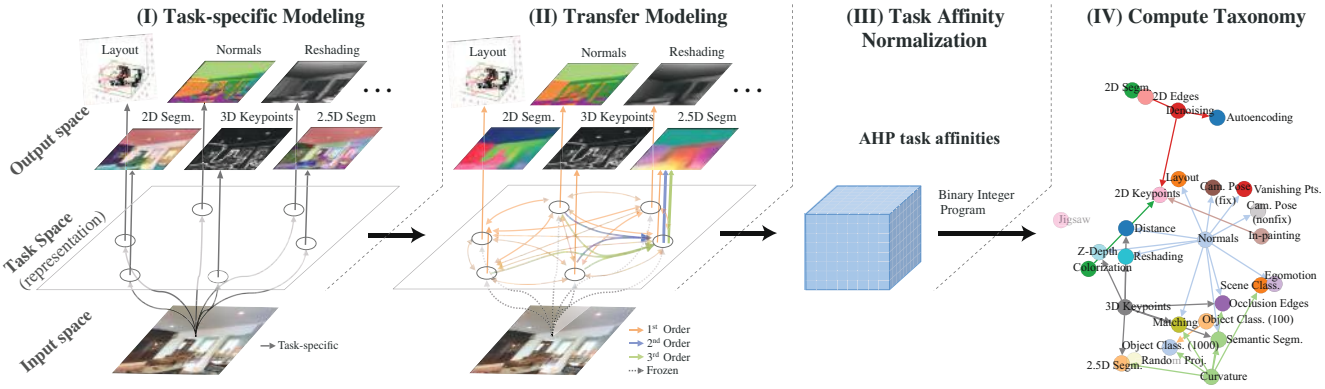


Figure 2: **Computational modeling of task relations and creating the taxonomy.** From left to right: I. Train task-specific networks. II. Train (first order and higher) transfer functions among tasks in a latent space. III. Get normalized transfer affinities using AHP (Analytic Hierarchy Process). IV. Find global transfer taxonomy using BIP (Binary Integer Program).

we want solved but cannot train ("target-only"), $\mathcal{T} \cap \mathcal{S}$ are the tasks that we want solved but could play as source too, and $\mathcal{S} - \mathcal{T} \cap \mathcal{S}$ are the "source-only" tasks which we may not directly care about to solve (e.g. jigsaw puzzle) but can be optionally used if they increase the performance on $\mathcal{T}$.

The **task** tax**onomy** (***taskonomy***) is a computationally found directed hypergraph that captures the notion of task transferability over any given task dictionary. An edge between a group of source tasks and a target task represents a feasible transfer case and its weight is the prediction of its performance. We use these edges to estimate the globally optimal transfer policy to solve $\mathcal{T}$. Taxonomy produces a family of such graphs, parameterized by the available supervision budget, chosen tasks, transfer orders, and transfer functions' expressiveness.

Taxonomy is built using a four step process depicted in Fig. 2. In stage I, a task-specific network for each task in $\mathcal{S}$ is trained. In stage II, all feasible transfers between sources and targets are trained. We include higher-order transfers which use multiple inputs task to transfer to one target. In stage III, the task affinities acquired from transfer function performances are normalized, and in stage IV, we synthesize a hypergraph which can predict the performance of any transfer policy and optimize for the optimal one.

A vision task is an abstraction read from a raw image. We denote a task t more formally as a function f_t which maps image I to $f_t(I)$. Our dataset, $\mathcal{D}$, contains for each task t a set of training pairs $(I, f_t(I))$, e.g. $(image, depth)$.

Task Dictionary: Our mapping of task space is done via (26) tasks included in the dictionary, so we ensure they cover common themes in computer vision (2D, 3D, semantics, etc) to the elucidate fine-grained structures of task space. See Fig. 3 for some of the tasks with detailed definition of each task provided in the supplementary material.

It is critical to note the task dictionary is meant to be a *sampled set, not an exhaustive list*, from a denser space of all conceivable visual tasks. This gives us a tractable way to sparsely model a dense space, and the hypothesis is that (subject to a proper sampling) the derived model should

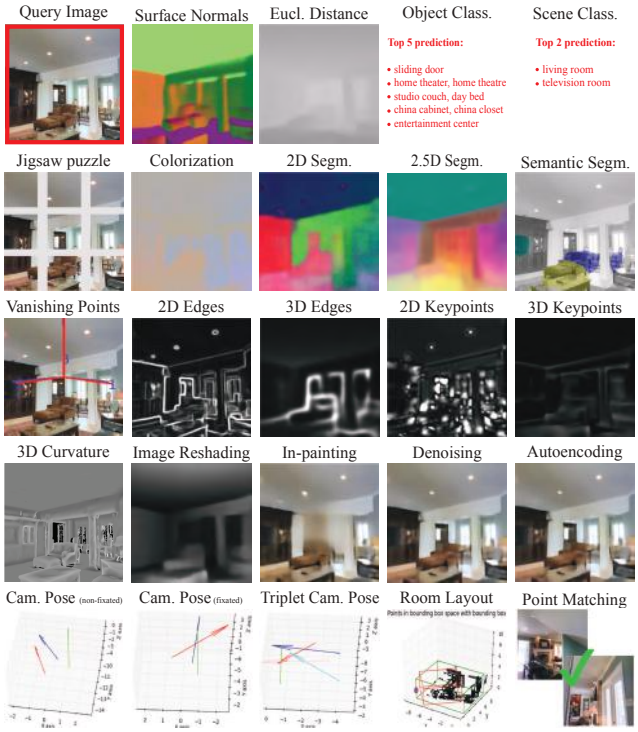


Figure 3: **Task Dictionary.** Outputs of 24 (of 26) task-specific networks for a query (top left). See results of applying frame-wise on a video here.

generalize to out-of-dictionary tasks. The more regular / better sampled the space, the better the generalization. We evaluate this in Sec. 4.2 with supportive results. For evaluation of the robustness of results w.r.t the choice of dictionary, see the supplementary material.

Dataset: We need a dataset that has annotations for *every task on every image*. Training all of our tasks on exactly the same pixels eliminates the possibility that the observed transferabilities are affected by different input data peculiarities rather than only task intrinsics. We created a dataset of 4 million images of indoor scenes from about 600 buildings; every image has an annotation for every task. The images are registered on and aligned with building-wide

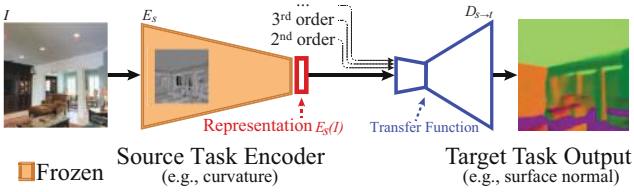


Figure 4: **Transfer Function**. A small readout function is trained to map representations of source task's frozen encoder to target task's labels. If order> 1, transfer function receives representations from multiple sources.

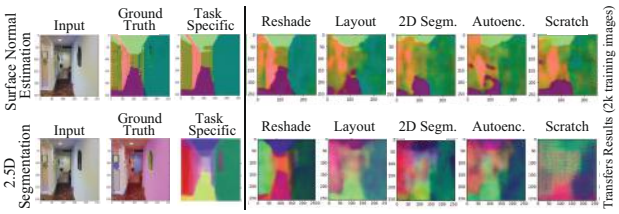


Figure 5: **Transfer results** to normals and 2.5D Segmentation from 5 different source tasks. The spread in transferability among sources is apparent. "Scratch" was trained from scratch without transfer learning.

meshes similar to [3, 98, 12] enabling us to programmatically compute the ground truth for many tasks without human labeling. For the tasks that still require labels (e.g. scene classes), we generate them using Knowledge Distillation [41] from known methods [101, 55, 54, 75]. See the supplementary material for full details of the process and a user study on the final quality of labels generated using Knowledge Distillation (showing $< 7\%$ error).

3.1. Step I: Task-Specific Modeling

We train a fully supervised task-specific network for each task in $\mathcal{S}$. Task-specific networks have an encoder-decoder architecture homogeneous across all tasks, where the encoder is large enough to extract powerful representations, and the decoder is large enough to achieve a good performance but is much smaller than the encoder.

3.2. Step II: Transfer Modeling

Given a source task s and a target task t, where $s \in \mathcal{S}$ and $t \in \mathcal{T}$, a transfer network learns a small readout function for t given a statistic computed for s (see Fig 4). The statistic is the representation for image I from the encoder of s: $E_s(I)$. The readout function ($D_{s\to t}$) is parameterized by $\theta_{s\to t}$ minimizing the loss L_t:

$$D_{s\to t} := \arg\min_{\theta} \mathbb{E}_{I\in\mathcal{D}}\Big[L_t\Big(D_\theta\big(E_s(I)\big), f_t(I)\Big)\Big], \quad (1)$$

where $f_t(I)$ is ground truth of t for image I. $E_s(I)$ may or may not be sufficient for solving t depending on the relation between t and s (examples in Fig. 5). Thus, the performance of $D_{s\to t}$ is a useful metric as task affinity. We train transfer functions for all feasible source-target combinations.

Accessibility: For a transfer to be successful, the latent representation of the source should both be *inclusive* of sufficient information for solving the target and have the information *accessible*, i.e. easily extractable (otherwise, the raw image or its compression based representations would be optimal). Thus, it is crucial for us to adopt a low-capacity (small) architecture as transfer function trained with a small amount of data, in order to measure transferability conditioned on being highly accessible. We use a shallow fully convolutional network and train it with little data (8x to 120x less than task-specific networks).

Higher-Order Transfers: Multiple source tasks can contain complementary information for solving a target task (see examples in Fig 6). We include higher-order transfers which are the same as first order but receive multiple representations in the input. Thus, our transfers are functions $D : \wp(\mathcal{S}) \to \mathcal{T}$, where $\wp$ is the powerset operator.

As there is a combinatorial explosion in the number of feasible higher-order transfers ($|\mathcal{T}| \times \binom{|\mathcal{S}|}{k}$ for k^{th} order), we employ a sampling procedure with the goal of filtering out higher-order transfers that are less likely to yield good results, without training them. We use a beam search: for transfers of order $k \leq 5$ to a target, we select its 5 best sources (according to 1^{st} order performances) and include all of their order-k combination. For $k \geq 5$, we use a beam of size 1 and compute the transfer from the top k sources.

We also tested transitive transfers ($s \to t_1 \to t_2$) which showed they do not improve the results, and thus, were not include in our model (results in supplementary material).

3.3. Step III: Ordinal Normalization using Analytic Hierarchy Process (AHP)

We want to have an affinity matrix of transferabilities across tasks. Aggregating the raw losses/evaluations $L_{s\to t}$ from transfer functions into a matrix is obviously problematic as they have vastly different scales and live in different spaces (see Fig. 7-left). Hence, a proper normalization is needed. A naive solution would be to linearly rescale each row of the matrix to the range $[0, 1]$. This approach fails when the actual output quality increases at different speeds w.r.t. the loss. As the loss-quality curve is generally unknown, such approaches to normalization are ineffective.

Instead, we use an *ordinal* approach in which the output quality and loss are only assumed to change monotonically. For each t, we construct W_t a pairwise tournament matrix between all feasible sources for transferring to t. The element at (i, j) is the percentage of images in a held-out test set, $\mathcal{D}_{test}$, on which s_i transfered to t better than s_j did (i.e. $D_{s_i\to t}(I) > D_{s_j\to t}(I)$).

We clip this intermediate pairwise matrix W_t to be in $[0.001, 0.999]$ as a form of Laplace smoothing. Then we divide $W'_t = W_t/W_t^T$ so that the matrix shows how many times better s_i is compared to s_j. The final tournament ratio matrix is positive reciprocal with each element $w'_{i,j}$ of W'_t:

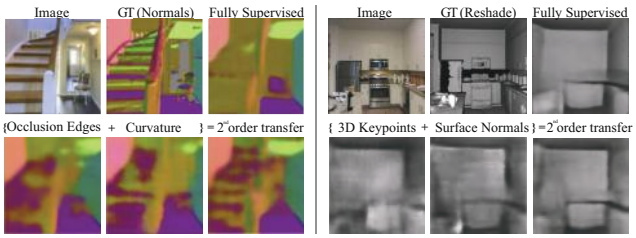


Figure 6: **Higher-Order Transfers.** Representations can contain complementary information. E.g. by transferring simultaneously from 3D Edges and Curvature individual stairs were brought out. See our publicly available interactive transfer visualization page for more examples.

$$w'_{i,j} = \frac{\mathbb{E}_{I \in \mathcal{D}_{test}}[D_{s_i \to t}(I) > D_{s_j \to t}(I)]}{\mathbb{E}_{I \in \mathcal{D}_{test}}[D_{s_i \to t}(I) < D_{s_j \to t}(I)]}. \quad (2)$$

We quantify the final transferability of s_i to t as the corresponding (i^{th}) component of the principal eigenvector of W'_t (normalized to sum to 1). The elements of the principal eigenvector are a measure of centrality, and are proportional to the amount of time that an infinite-length random walk on W'_t will spend at any given source [59]. We stack the principal eigenvectors of W'_t for all $t \in \mathcal{T}$, to get an affinity matrix P ('p' for performance)—see Fig. 7, right.

This approach is derived from Analytic Hierarchy Process [76], a method widely used in operations research to create a total order based on multiple pairwise comparisons.

3.4. Step IV: Computing the Global Taxonomy

Given the normalized task affinity matrix, we need to devise a global transfer policy which maximizes collective performance across all tasks, while minimizing the used supervision. This problem can be formulated as subgraph selection where tasks are nodes and transfers are edges. The optimal subgraph picks the ideal source nodes and the best edges from these sources to targets while satisfying that the number of source nodes does not exceed the supervision budget. We solve this subgraph selection problem using Boolean Integer Programming (BIP), described below, which can be solved optimally and efficiently [39, 14].

Our transfers (edges), E, are indexed by i with the form $(\{s^i_1, \ldots, s^i_{m_i}\}, t^i)$ where $\{s^i_1, \ldots, s^i_{m_i}\} \subset \mathcal{S}$ and $t^i \in \mathcal{T}$. We define operators returning target and sources of an edge:

$$\big(\{s^i_1, \ldots, s^i_{m_i}\}, t^i\big) \xmapsto{sources} \{s^i_1, \ldots, s^i_{m_i}\}$$
$$\big(\{s^i_1, \ldots, s^i_{m_i}\}, t^i\big) \xmapsto{target} t^i.$$

Solving a task t by fully supervising it is denoted as $\big(\{t\}, t\big)$. We also index the targets $\mathcal{T}$ with j so that in this section, i is an edge and j is a target.

The parameters of the problem are: the supervision budget (γ) and a measure of performance on a target from each of its transfers (p_i), i.e. the affinities from P. We can also optionally include additional parameters of: r_j specifying the relative importance of each target task and ℓ_i specifying the relative cost of acquiring *labels* for each task.

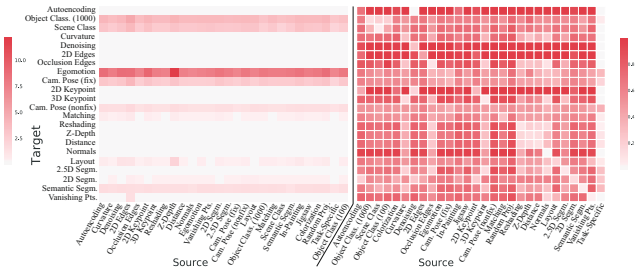


Figure 7: **First-order task affinity matrix** before (left) and after (right) Analytic Hierarchy Process (AHP) normalization. Lower means better transfered. For visualization, we use standard affinity-distance method $dist = e^{-\beta \cdot P}$ (where $\beta = 20$ and e is element-wise matrix exponential). See supplementary material for the full matrix with higher-order transfers.

The BIP is parameterized by a vector x where each transfer and each task is represented by a binary variable; x indicates which nodes are picked to be source and which transfers are selected. The canonical form for a BIP is:

$$\text{maximize } c^T x\,,$$
$$\text{subject to } Ax \preceq b$$
$$\text{and } x \in \{0,1\}^{|E|+|\mathcal{V}|}\,.$$

Each element c_i for a transfer is the product of the importance of its target task and its transfer performance:

$$c_i := r_{target(i)} \cdot p_i\,. \quad (3)$$

Hence, the *collective* performance on all targets is the summation of their individual AHP performance, p_i, weighted by the user specified importance, r_i.

Now we add three types of constraints via matrix A to enforce each feasible solution of the BIP instance corresponds to a valid subgraph for our transfer learning problem: *Constraint I:* if a transfer is included in the subgraph, all of its source nodes/tasks must be included too, *Constraint II:* each target task has exactly one transfer in, *Constraint III:* supervision budget is not exceeded.

Constraint I: For each row a_i in A we require $a_i \cdot x \leq b_i$, where

$$a_{i,k} = \begin{cases} |sources(i)| & \text{if } k = i \\ -1 & \text{if } (k - |E|) \in sources(i) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$b_i = 0. \quad (5)$$

Constraint II: Via the row $a_{|E|+j}$, we enforce that each target has exactly one transfer:

$$a_{|E|+j,i} := 2 \cdot \mathbb{1}_{\{target(i)=j\}}, \qquad b_{|E|+j} := -1. \quad (6)$$

Constraint III: the solution is enforced to not exceed the budget. Each transfer i is assigned a label cost ℓ_i, so

$$a_{|E|+|\mathcal{V}|+1,i} := \ell_i, \qquad b_{|E|+|\mathcal{V}|+1} := \gamma. \quad (7)$$

The elements of A not defined above are set to 0. The problem is now a valid BIP and can be optimally solved in a fraction of a second [39]. The BIP solution $\hat{x}$ corresponds to the optimal subgraph, which is our taxonomy.

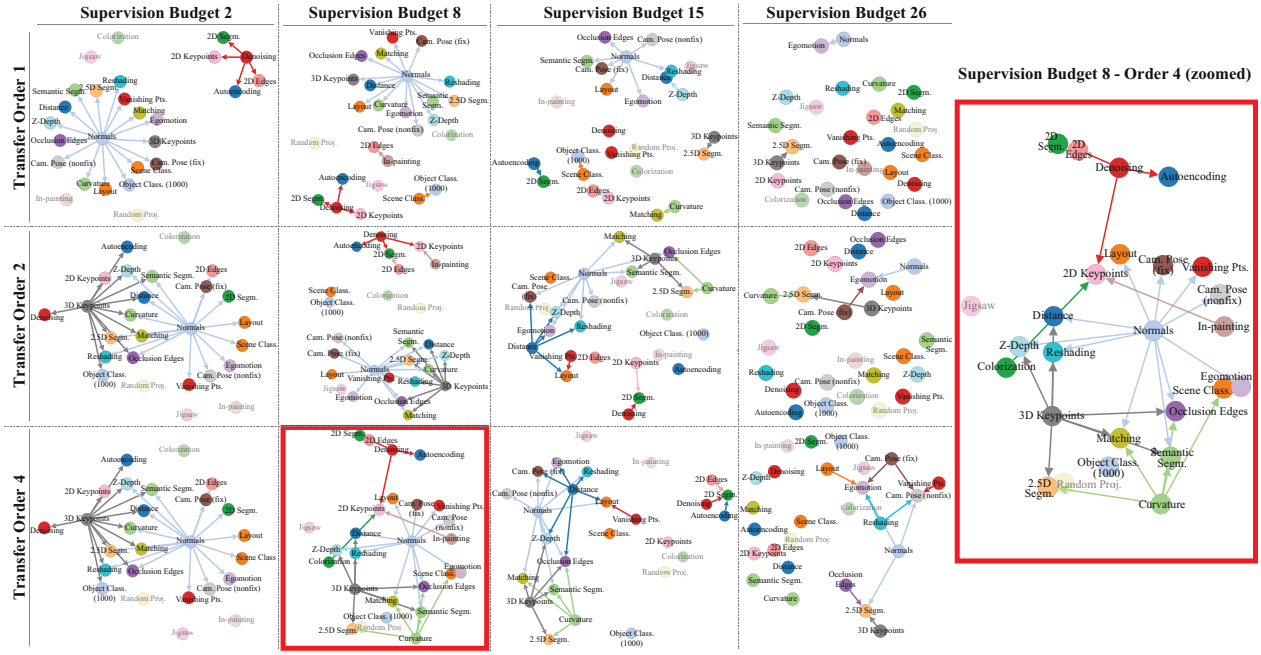


Figure 8: **Computed taxonomies** for solving 22 tasks given various supervision budgets (x-axes), and maximum allowed transfer orders (y-axes). One is magnified for better visibility. Nodes with incoming edges are target tasks, and the number of their incoming edges is the order of their chosen transfer function. Still transferring to some targets when tge budget is 26 (full budget) means certain transfers started performing better than their fully supervised task-specific counterpart. See the interactive solver website for color coding of the nodes by *Gain* and *Quality* metrics. Dimmed nodes are the source-only tasks, and thus, only participate in the taxonomy if found worthwhile by the BIP optimization to be one of the sources.

4. Experiments

With 26 tasks in the dictionary (4 source-only tasks), our approach leads to training 26 fully supervised task-specific networks, 22×25 transfer networks in 1^{st} order, and $22 \times \binom{25}{k}$ for k^{th} order, from which we sample according to the procedure in Sec. 3. The total number of transfer functions trained for the taxonomy was $\sim$3,000 which took 47,886 GPU hours on the cloud.

Out of 26 tasks, we usually use the following 4 as source-only tasks (described in Sec. 3) in the experiments: colorization, jigsaw puzzle, in-painting, random projection. However, the method is applicable to an arbitrary partitioning of the dictionary into $\mathcal{T}$ and $\mathcal{S}$. The interactive solver website allows the user to specify any desired partition.

Network Architectures: We preserved the architectural and training details across tasks as homogeneously as possible to avoid injecting any bias. The **encoder** architecture is identical across all task-specific networks and is a fully convolutional ResNet-50 without pooling. All **transfer** functions include identical shallow networks with 2 conv layers (concatenated channel-wise if higher-order). The loss (L_t) and **decoder**'s architecture, though, have to depend on the task as the output structures of different tasks vary; for all pixel-to-pixel tasks, e.g. normal estimation, the decoder is a 15-layer fully convolutional network; for low dimensional tasks, e.g. vanishing points, it consists of 2-3 FC layers. All networks are trained using the same hyperparameters regardless of task and on exactly the same input images. Tasks with more than one input, e.g. relative camera pose, share weights between the encoder towers. Transfer networks are all trained using the same hyperparameters as the task-specific networks, except that we anneal the learning rate earlier since they train much faster. Detailed definitions of architectures, training process, and experiments with different encoders can be found in the supplementary material.

Data Splits: Our dataset includes 4 million images. We made publicly available the models trained on full dataset, but for the experiments reported in the main paper, we used a subset of the dataset as the extracted structure stabilized and did not change when using more data (explained in Sec. 5.2). The used subset is partitioned into training (120k), validation (16k), and test (17k) images, each from non-overlapping sets of buildings. Our task-specific networks are trained on the training set and the transfer networks are trained on a subset of validation set, ranging from 1k images to 16k, in order to model the transfer patterns under different data regimes. In the main paper, we report all results under the 16k transfer supervision regime ($\sim$10% of the split) and defer the additional sizes to the supplementary material and website (see Sec. 5.2). Transfer functions are evaluated on the test set.

How good are the trained task-specific networks? *Win rate (%)* is the proportion of test set images for which a baseline is beaten. Table 1 provides win rates of the task-specifc networks vs. two baselines. Visual outputs for a random test sample are in Fig. 3. The high win rates in Table 1 and qualitative results show the networks are well trained and stable and can be relied upon for modeling the task space. See results of applying the networks on a YouTube video frame-by-frame here. A live demo for user uploaded queries is available here.

Task	*avg*	*rand*	Task	*avg*	*rand*	Task	*avg*	*rand*
Denoising	100	99.9	Layout	99.6	89.1	Scene Class.	97.0	93.4
Autoenc.	100	99.8	2D Edges	100	99.9	Occ. Edges	100	95.4
Reshading	94.9	95.2	Pose (fix)	76.3	79.5	Pose (nonfix)	60.2	61.9
Inpainting	99.9	-	2D Segm.	97.7	95.7	2.5D Segm.	94.2	89.4
Curvature	78.7	93.4	Matching	86.8	84.6	Egomotion	67.5	72.3
Normals	99.4	99.5	Vanishing	99.5	96.4	2D Keypnt.	99.8	99.4
Z-Depth	92.3	91.1	Distance	92.4	92.1	3D Keypnt.	96.0	96.9
Mean	92.4	90.9						

Table 1: **Task-Specific Networks' Sanity:** Win rates vs. *random* (Gaussian) network representation readout and statistically informed guess *avg*.

To get a sense of the quality of our networks vs. state-of-the-art task-specific methods, we compared our depth estimator vs. released models of [51] which led to outperforming [51] with a win rate of 88% and losses of 0.35 vs. 0.47 (further details in the supplementary material). In general, we found the task-specific networks to perform on par or better than state-of-the-art for many of the tasks, though we do not formally benchmark or claim this.

4.1. Evaluation of Computed Taxonomies

Fig. 8 shows the computed taxonomies optimized to solve the full dictionary, i.e. all tasks are placed in $\mathcal{T}$ and $\mathcal{S}$ (except for 4 source-only tasks that are in $\mathcal{S}$ only). This was done for various supervision budgets (columns) and maximum allowed order (rows) constraints. Still seeing transfers to some targets when the budget is 26 (full dictionary) means certain transfers became better than their fully supervised task-specific counterpart.

While Fig. 8 shows the structure and connectivity, Fig. 9 quantifies the results of taxonomy recommended transfer policies by two metrics of *Gain* and *Quality*, defined as:

Gain: win rate (%) against a network trained from scratch using the same training data as transfer networks'. That is, the best that could be done if transfer learning was not utilized. This quantifies the *gained* value by transferring.

Quality: win rate (%) against a fully supervised network trained with 120k images (gold standard).

Each column in Fig. 9 shows a supervision budget. As apparent, good results can be achieved even when the supervision budget is notably smaller than the number of solved tasks, and as the budget increases, results improve (expected). Results are shown for 2 maximum allowed orders.

4.2. Generalization to Novel Tasks

The taxonomies in Sec. 4.1 were optimized for solving all tasks in the dictionary. In many situations, a practitioner is interested in a single task which even may not be in the dictionary. Here we evaluate how taxonomy transfers to a novel out-of-dictionary task with little data.

This is done in an all-for-one scenario where we put one task in $\mathcal{T}$ and all others in $\mathcal{S}$. The task in $\mathcal{T}$ is target-only and has no task-specific network. Its limited data (16k) is used to train small transfer networks to sources. This basically *localizes* where the target would be in the taxonomy.

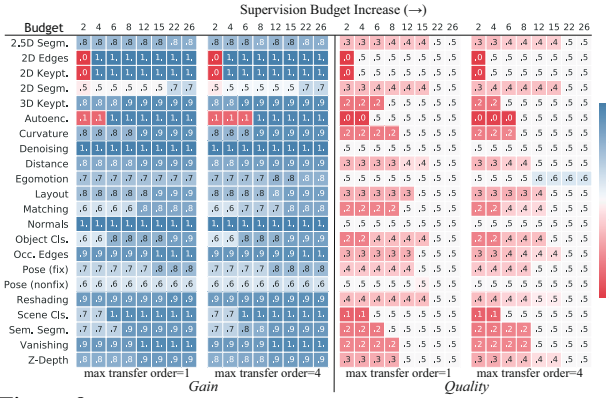
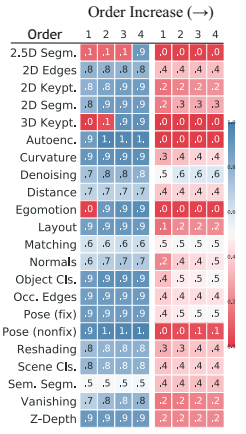


Figure 9: **Evaluation of taxonomy computed for solving the full task dictionary**. Gain (left) and Quality (right) values for each task using the policy suggested by the computed taxonomy, as the supervision budget increases(→). Shown for transfer orders 1 and 4.



Task	scratch	ImageNet[49]	Wang[93]	Agrawal[1]	Zamir[97]	Zhang[100]	Noroozi[65]	full sup.	Taxonomy
Depth	88	88	93	89	88	84	86	43	-
	.03	.04	.04	.03	.04	.03	.03	**.02**	.02
Scene Cls.	80	52	83	74	74	71	75	15	-
	3.30	2.76	3.56	3.15	3.17	3.09	3.19	**2.23**	2.63
Sem. Segm.	78	79	82	85	76	78	84	21	-
	1.74	1.88	1.92	1.80	1.85	1.74	1.71	**1.42**	1.53
Object Cls.	79	54	82	76	75	76	76	34	-
	4.08	3.57	4.27	3.99	3.98	4.00	3.97	**3.26**	3.46
Normals	97	98	98	98	98	97	97	6	-
	.22	.30	.34	.28	.28	.23	.24	**.12**	.15
2.5D Segm.	80	93	92	89	90	84	87	40	-
	.21	.34	.34	.26	.29	.22	.24	**.16**	.17
Occ. Edges	93	96	95	93	94	93	94	42	-
	.16	.19	.18	.17	.18	.16	.17	**.12**	.13
Curvature	88	94	89	85	88	92	88	29	-
	.25	.28	.26	.25	.26	.26	.25	**.21**	.22
Egomotion	79	78	83	77	76	74	71	59	-
	8.60	8.58	9.26	8.41	8.34	8.15	7.94	7.32	**6.85**
Layout	80	76	85	79	77	78	70	36	-
	.66	.66	.85	.65	.65	.62	.54	**.37**	.41

Figure 10: **Generalization to Novel Tasks**. Each row shows a novel test task. Left: Gain and Quality values using the devised "all-for-one" transfer policies for novel tasks for orders 1-4. Right: Win rates (%) of the transfer policy over various self-supervised methods, ImageNet features, and scratch are shown in the colored rows. Note the large margin of win by taxonomy. The uncolored rows show corresponding loss values.

Fig. 10 (left) shows the *Gain* and *Quality* of the transfer policy found by the BIP for each task. Fig. 10 (right) compares the taxonomy suggested policy against some of the best existing self-supervised methods [93, 100, 65, 97, 1], ImageNet FC7 features [49], training from scratch, and a fully supervised network (gold standard).

The results in Fig. 10 (right) are noteworthy. The large win margin for taxonomy shows that carefully selecting transfer policies depending on the target is superior to fixed transfers, such as the ones employed by self-supervised methods. ImageNet features which are the most popular off-the-shelf features in vision are also outperformed by those policies. Additionally, though the taxonomy transfer policies lose to fully supervised networks (gold standard) in most cases, the results often get close with win rates in 40% range. These observations suggests the space has a rather predicable and strong structure. For graph visualization of

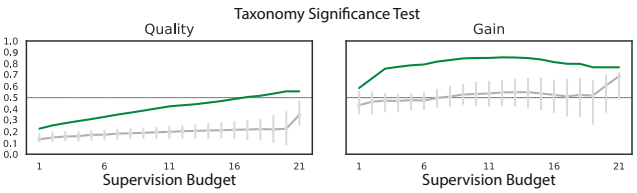


Figure 11: **Structure Significance.** Our taxonomy compared with random transfer policies (random feasible taxonomies that use the maximum allowable supervision budget). Y-axis shows *Quality* or *Gain*, and X-axis is the supervision budget. Green and gray represent our taxonomy and random connectivities, respectively. Error bars denote 5^{th}–95^{th} percentiles.

the all-for-one taxonomy policies please see the supplementary material. The solver website allows generating the taxonomy for arbitrary sets of target-only tasks.

5. Significance Test of the Structure

The previous evaluations showed good transfer results in terms of *Quality* and *Gain*, but how crucial is it to use our taxonomy to choose smart transfers over just choosing *any* transfer? In other words, how *significant/strong* is the discovered structure of task space? Fig. 11 quantifies this by showing the performance of our taxonomy versus a large set of taxonomies with random connectivities. Taxonomy outperformed all other connectivities by a large margin signifying both existence of a strong structure in the space as well as a good modeling of it by our approach. See the supplementary material for full experimental details.

5.1. Evaluation on MIT Places & ImageNet

To what extent are our findings dataset dependent, and would the taxonomy change if done on another dataset? We examined this by finding the ranking of all tasks for transferring to two target tasks of object classification and scene classification on our dataset. We then fine tuned our task-specific networks on other datasets (MIT Places [101] for scene classification, ImageNet [75] for object classification) and evaluated them on their respective test sets and metrics. Fig. 12 shows how the results correlate with taxonomy's ranking from our dataset. The Spearman's rho between the taxonomy ranking and the Top-1 ranking is 0.857 on Places and 0.823 on ImageNet showing a notable correlation. See the supplementary material for full experimental details.

5.2. Universality of the Structure

We employed a computational approach with various design choices. It is important to investigate how specific to those the discovered structure is. We did stability tests by computing the variance in our output when making changes in one of the following system choices: **I. architecture of task-specific networks, II. architecture of transfer function networks, III. amount of data available for training transfer networks, IV. datasets, V. data splits, VI. choice of dictionary**. Overall, despite injecting large changes (e.g. varying the size of training data of transfer functions by 16x,

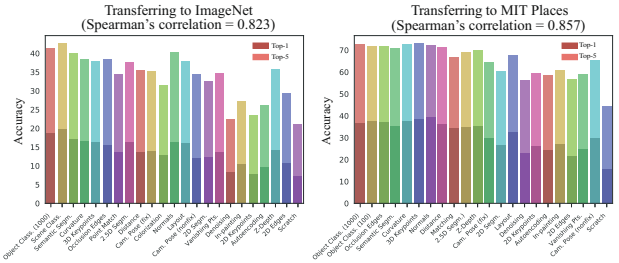


Figure 12: **Evaluating the discovered structure on other datasets: ImageNet [75] and MIT Places [101].** Y-axis shows accuracy on the external benchmark while bars on x-axis are ordered by taxonomy's predicted performance based on our dataset. A monotonically decreasing plot corresponds to preserving identical orders and perfect generalization.

size and architecture of task-specific networks and transfer networks by 4x), we found the outputs to be remarkably stable leading to almost no change in the output taxonomy computed on top. Detailed results and experimental setup of each tests are reported in the supplementary material.

6. Limitations and Discussion

We presented a method for modeling the space of visual tasks by way of transfer learning and showed its utility in reducing the need for supervision. The space of tasks is an interesting object of study in its own right and we have only scratched the surface in this regard. We also made a number of assumptions in the framework which should be noted.

Model Dependence: We used a computational approach and adopted neural networks as our function class. Though we validated the stability of the findings w.r.t various architectures and datasets, it should be noted that the findings are in principle model and data specific.

Compositionality: We performed the modeling via a set of common human-defined visual tasks. It is natural to consider a further compositional approach in which such common tasks are viewed as *observed samples* which are composed of computationally found latent subtasks.

Space Regularity: We performed modeling of a dense space via a sampled dictionary. Though we showed a good tolerance w.r.t. to the choice of dictionary and transferring to out-of-dictionary tasks, this outcome holds upon a proper sampling of the space as a function of its regularity. More formal studies on properties of the computed space is required for this to be provably guaranteed for a general case.

Lifelong Learning: We performed the modeling in one go. In many cases, e.g. lifelong learning, the system is evolving and the number of mastered tasks constantly increase. Such scenarios require augmentation of the structure with expansion mechanisms based on new beliefs.

Acknowledgement: We acknowledge the support of NSF (DMS-1521608), MURI (1186514-1-TBCJE), ONR MURI (N00014-14-1-0671), Toyota(1191689-1-UDAWF), ONR MURI (N00014-13-1-0341), Nvidia, Tencent, a gift by Amazon Web Services, a Google Focused Research Award.

References

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 37–45, 2015. 7

[2] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, and N. de Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pages 3981–3989, 2016. 2

[3] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 4

[4] S. Arora, A. Bhaskara, R. Ge, and T. Ma. Provable bounds for learning some deep representations. In *International Conference on Machine Learning*, pages 584–592, 2014. 1

[5] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2252–2259. IEEE, 2011. 2

[6] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 2

[7] P. Berkhin et al. A survey of clustering data mining techniques. *Grouping multidimensional data*, 25:71, 2006. 2

[8] E. Bienenstock, S. Geman, and D. Potter. Compositionality, mdl priors, and object recognition. In *Advances in neural information processing systems*, pages 838–844, 1997. 2

[9] H. Bilen and A. Vedaldi. Integrated perception with recurrent multi-task neural networks. In *Advances in neural information processing systems*, pages 235–243, 2016. 2

[10] J. Bingel and A. Søgaard. Identifying beneficial task relations for multi-task learning in deep neural networks. *arXiv preprint arXiv:1702.08303*, 2017. 2

[11] O. Boiman and M. Irani. Similarity by composition. In *Advances in neural information processing systems*, pages 177–184, 2007. 2

[12] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 4

[13] Z. Chen and B. Liu. *Lifelong Machine Learning*. Morgan & Claypool Publishers, 2016. 2

[14] I. I. CPLEX. V12. 1: Users manual for cplex. *International Business Machines Corporation*, 46(53):157, 2009. 5

[15] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015. 2

[16] C. Doersch and A. Zisserman. Multi-task self-supervised visual learning. *arXiv preprint arXiv:1708.07860*, 2017. 2

[17] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014. 2

[18] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 2

[19] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel. Rl2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016. 2

[20] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010. 2

[21] A. Faktor and M. Irani. clustering by composition–unsupervised discovery of image categories. In *European Conference on Computer Vision*, pages 474–487. Springer, 2012. 2

[22] L. Fe-Fei et al. A bayesian approach to unsupervised one-shot learning of object categories. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1134–1141. IEEE, 2003. 2

[23] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006. 2

[24] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013. 2

[25] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017. 2

[26] C. Finn, S. Levine, and P. Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. *CoRR*, abs/1603.00448, 2016. 2

[27] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel. Deep spatial autoencoders for visuomotor learning. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 512–519. IEEE, 2016. 2

[28] C. Finn, T. Yu, J. Fu, P. Abbeel, and S. Levine. Generalizing skills with semi-supervised reinforcement learning. *CoRR*, abs/1612.00429, 2016. 2

[29] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine. One-shot visual imitation learning via meta-learning. *CoRR*, abs/1709.04905, 2017. 2

[30] I. K. Fodor. A survey of dimension reduction techniques. Technical report, Lawrence Livermore National Lab., CA (US), 2002. 2

[31] R. M. French. Catastrophic forgetting in connectionist networks: Causes, consequences and solutions. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. 2

[32] R. Ge. *Provable algorithms for machine learning problems*. PhD thesis, Princeton University, 2013. 1

[33] S. Geman, D. F. Potter, and Z. Chi. Composition systems. *Quarterly of Applied Mathematics*, 60(4):707–736, 2002. 2

[34] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 999–1006. IEEE, 2011. 2

[35] A. Gopnik, C. Glymour, D. Sobel, L. Schulz, T. Kushnir, and D. Danks. A theory of causal learning in children: Causal maps and bayes nets. 111:3–32, 02 2004. 2

[36] A. Gopnik, C. Glymour, D. M. Sobel, L. E. Schulz, T. Kushnir, and D. Danks. A theory of causal learning in

children: causal maps and bayes nets. *Psychological review*, 111(1):3, 2004. 2

[37] A. Gopnik, A. N. Meltzoff, and P. K. Kuhl. *The scientist in the crib: Minds, brains, and how children learn.* William Morrow & Co, 1999. 2

[38] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *CoRR*, abs/1410.5401, 2014. 2

[39] I. Gurobi Optimization. Gurobi optimizer reference manual, 2016. 5

[40] K. Henry. The theory and applications of homomorphic cryptography. Master's thesis, University of Waterloo, 2008. 2

[41] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 4

[42] J. Hoffman, T. Darrell, and K. Saenko. Continuous manifold based adaptation for evolving visual domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 867–874, 2014. 2

[43] Y. Hoshen and S. Peleg. Visual learning of arithmetic operations. *CoRR*, abs/1506.02264, 2015. 2

[44] F. Hu, G.-S. Xia, J. Hu, and L. Zhang. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, 7(11):14680–14707, 2015. 2

[45] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang. Robust visual domain adaptation with low-rank reconstruction. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2168–2175. IEEE, 2012. 2

[46] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 2

[47] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[48] I. Kokkinos. Ubernet: Training auniversal'convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. *arXiv preprint arXiv:1609.02132*, 2016. 2

[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 7

[50] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1785–1792. IEEE, 2011. 2

[51] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016. 7

[52] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 2

[53] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, pages 1–101, 2016. 2

[54] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. *arXiv preprint arXiv:1611.07709*, 2016. 4

[55] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4

[56] F. Liu, G. Lin, and C. Shen. CRF learning with CNN features for image segmentation. *CoRR*, abs/1503.08263, 2015. 2

[57] Z. Luo, Y. Zou, J. Hoffman, and L. Fei-Fei. Label efficient learning of transferable representations across domains and tasks. 2

[58] J. Malik, P. Arbeláez, J. Carreira, K. Fragkiadaki, R. Girshick, G. Gkioxari, S. Gupta, B. Hariharan, A. Kar, and S. Tulsiani. The three rs of computer vision: Recognition, reconstruction and reorganization. *Pattern Recognition Letters*, 72:4–14, 2016. 2

[59] N. Masuda, M. A. Porter, and R. Lambiotte. Random walks and diffusion on networks. *Physics Reports*, 716-717:1 – 58, 2017. Random walks and diffusion on networks. 5

[60] M. Mccloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *The Psychology of Learning and Motivation*, 24:104–169, 1989. 2

[61] L. Mihalkova, T. Huynh, and R. J. Mooney. Mapping and revising markov logic networks for transfer learning. In *AAAI*, volume 7, pages 608–614, 2007. 2

[62] T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168, 2013. 2

[63] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016. 2

[64] A. Niculescu-Mizil and R. Caruana. Inductive transfer for bayesian network structure learning. In *Artificial Intelligence and Statistics*, pages 339–346, 2007. 2

[65] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016. 2, 7

[66] M. Noroozi, H. Pirsiavash, and P. Favaro. Representation learning by learning to count. *arXiv preprint arXiv:1708.06734*, 2017. 2

[67] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013. 2

[68] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30, 2012. 2

[69] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 2

[70] A. Pentina and C. H. Lampert. Multi-task learning with labeled and unlabeled tasks. *stat*, 1050:1, 2017. 2

[71] J. Piaget and M. Cook. *The origins of intelligence in children*, volume 8. International Universities Press New York, 1952. 2

[72] L. Y. Pratt. Discriminability-based transfer between neural networks. In *Advances in neural information processing systems*, pages 204–211, 1993. 2

[73] S. R. Richter, Z. Hayder, and V. Koltun. Playing for benchmarks. In *International Conference on Computer Vision (ICCV)*, 2017. 2

[74] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000. 2

[75] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 2, 4, 8

[76] R. W. Saaty. The analytic hierarchy process – what it is and how it is used. *Mathematical Modeling*, 9(3-5):161–176, 1987. Mat/d Modelling, Vol. 9, No. 3-5, pp. 161-176, 1987. 5

[77] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. *Computer Vision–ECCV 2010*, pages 213–226, 2010. 2

[78] R. Salakhutdinov, J. Tenenbaum, and A. Torralba. One-shot learning with a hierarchical nonparametric bayesian model. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 195–206, 2012. 2

[79] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel. Trust region policy optimization. *CoRR*, abs/1502.05477, 2015. 2

[80] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014. 2

[81] D. L. Silver and K. P. Bennett. Guest editors introduction: special issue on inductive transfer learning. *Machine Learning*, 73(3):215–220, 2008. 2

[82] D. L. Silver, Q. Yang, and L. Li. Lifelong machine learning systems: Beyond learning algorithms. In *in AAAI Spring Symposium Series*, 2013. 2

[83] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013. 2

[84] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. 2

[85] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013. 2

[86] J. B. Tenenbaum and T. L. Griffiths. Generalization, similarity, and bayesian inference. *Behavioral and Brain Sciences*, 24(4):629640, 2001. 2

[87] J. B. Tenenbaum, C. Kemp, T. L. Griffiths, and N. D. Goodman. How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022):1279–1285, 2011. 2

[88] J. B. Tenenbaum, C. Kemp, and P. Shafto. Theory-based bayesian models of inductive learning and reasoning. In *Trends in Cognitive Sciences*, pages 309–318, 2006. 2

[89] D. G. R. Tervo, J. B. Tenenbaum, and S. J. Gershman. Toward the neural implementation of structure learning. *Current opinion in neurobiology*, 37:99–105, 2016. 2

[90] C. Tessler, S. Givony, T. Zahavy, D. J. Mankowitz, and S. Mannor. A deep hierarchical approach to lifelong learning in minecraft. In *AAAI*, pages 1553–1561, 2017. 2

[91] S. Thrun and L. Pratt. *Learning to learn*. Springer Science & Business Media, 2012. 2

[92] A. M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950. 2

[93] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2015. 7

[94] X. Wang, K. He, and A. Gupta. Transitive invariance for self-supervised visual representation learning. *arXiv preprint arXiv:1708.02901*, 2017. 2

[95] T. Winograd. *Thinking machines: Can there be? Are we*, volume 200. University of California Press, Berkeley, 1991. 2

[96] J. Yang, R. Yan, and A. G. Hauptmann. Adapting svm classifiers to data with shifted distributions. In *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*, pages 69–76. IEEE, 2007. 2

[97] A. R. Zamir, T. Wekel, P. Agrawal, C. Wei, J. Malik, and S. Savarese. Generic 3d representation via pose estimation and matching. In *European Conference on Computer Vision*, pages 535–553. Springer, 2016. 2, 7

[98] A. R. Zamir, F. Xia, J. He, A. Sax, J. Malik, and S. Savarese. Gibson Env: Real-world perception for embodied agents. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 4

[99] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530, 2016. 2

[100] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016. 2, 7

[101] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014. 2, 4, 8