Taskonomy：Disentangling Task Transfer Learning

一、Background

人类的视觉具备多种多样的能力，计算机视觉届基于此定义了许多不同的视觉任务。长远来看，计算机视觉着眼于解决大多数甚至所有视觉任务，但现有方法大多尝试将视觉任务逐一击破。这种方法造成了两个问题：

1、逐一击破需要为每一项任务收集大量数据，随着任务数量的增多，这将会是不可行的；

2、逐一击破会带来不同任务之间的冗余计算和重复学习。总的来说，逐一击破的策略忽略了视觉任务之间的关联性，比如法线（Surface Normals）是由深度（Depth）求导得来，语义分割（Semantic Segmentation）又似乎和遮挡边缘测试（Occlusion edge detection）有着千丝万缕的关联。

总的来说，逐一击破的策略忽略了视觉任务之间的关联性，比如法线（Surface Normals）是由深度（Depth）求导得来，语义分割（Semantic Segmentation）又似乎和遮挡边缘测试（Occlusion edge detection）有着千丝万缕的关联。基于上述两个问题，我们希望能有效测量并利用视觉任务之间的关联来避免重复学习，从而用更少的数据学习我们感兴趣的一组任务。、逐一击破需要为每一项任务收集大量数据，随着任务数量的增多，这将会是不可行的；

二、Introduction

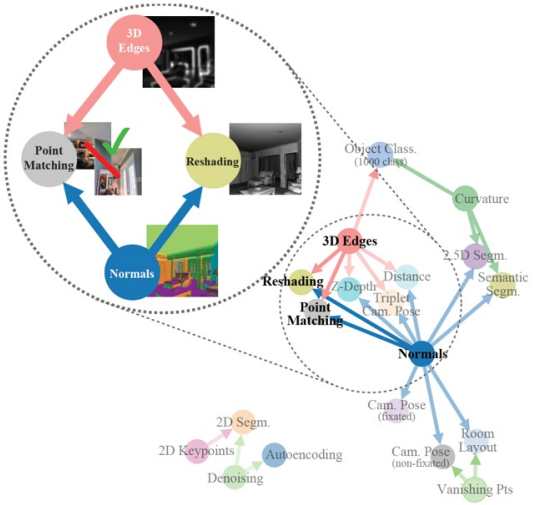


Figure 1: **A sample task structure discovered by the computational task taxonomy (*taskonomy*).** It found that, for instance, by combining the learned features of a surface normal estimator and occlusion edge detector, good networks for reshading and point matching can be rapidly trained with little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

Taskonomy 是一项**量化**不同视觉任务之间关联、并利用这些关联来最优化**学习策略**的研究。如果两个视觉任务 A、B 具有关联性，那么在任务 A 中习得的 representations 理应可为解决任务 B 提供有效的统计信息 。由此我们通过迁移学习计算了 26 个不同视觉任务之间的一阶以及高阶关联。如图一，如果有预测法线的网络和预测遮挡边缘测试的网络，我们可

以通过结合两个网络的 representations 来快速通过少量数据解决 Reshading 和点匹配 (Point matching)。基于这些关联，我们利用 BIP (Binary Integer Programming) 求得对于一组我们感兴趣的任务，如何去最优分配训练数据量。 比如，如果想最高效地解决 10 个问题，利用 Taskonomy 提供的学习策略可以减少 2/3 的训练数据量。

三、Method

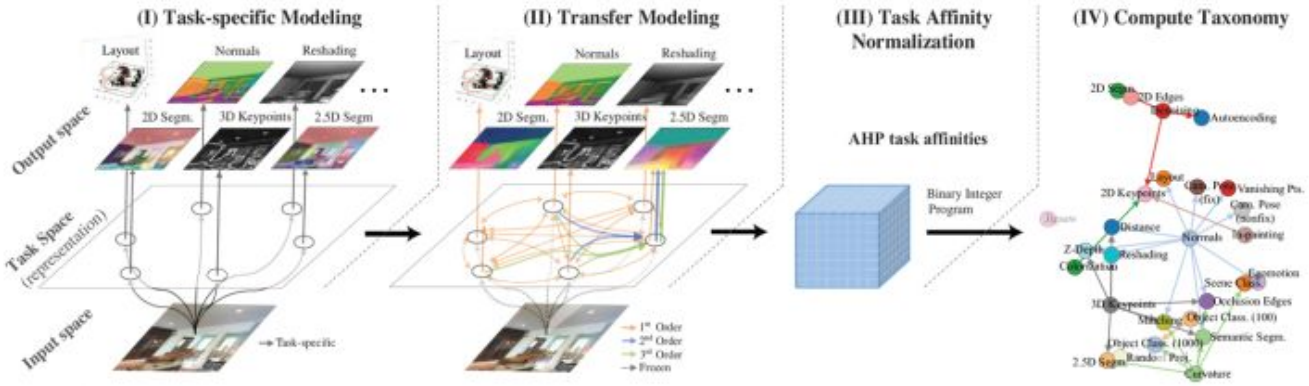


Figure 2: **Computational modeling of task relations and creating the taxonomy.** From left to right: I. Train task-specific networks. II. Train (first order and higher) transfer functions among tasks in a latent space. III. Get normalized transfer affinities using AHP (Analytic Hierarchy Process). IV.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

方法分为两个大阶段，四个小步。
第一大阶段涉及前三小步，我们要量化不同视觉任务之间的关联，并将任务关联表达成一个 affinity matrix（关联矩阵）。
第二大阶段，也就是最后一步，我们对求得的 affinity matrix 进行最优化，求得如何最高效地去学习一组任务。这个最高效的策略会由一个指向图 (directed graph) 来表示，我们称此指向图为 Taskonomy。词语上 Taskonomy 是 Task (任务) 和 Taxonomy (分类论) 的合并简称。

问题定义：

首先，我们来定义我们想要解决的问题。我们想在有限的**监督预算** γ 下最大化我们在一组**目标任务** (target tasks) $\mathcal{T} = \{t_1, \ldots, t_n\}$ 上的表现。同时，我们有一组**起始任务** (source tasks) $\mathcal{S}$，其定义为我们可从零学习的任务。监督预算 γ 的定义为多少起始任务我们愿意从零开始学习（从零开始学习需要收集大量数据，监督预算表达了我们所面对的金钱、计算力和时间上的限制）。那么，$\mathcal{T} \setminus \mathcal{S}$ 代表了我们感兴趣但不能从零学习的任务，比如一个只能有少量数据的任务。$\mathcal{S} \setminus \mathcal{T}$ 代表了我们不感兴趣但可以从零学习（来帮助我们更好的学习 $\mathcal{T}$ ）的任务，如jigsaw、colorization等自我监督的视觉任务。 $\mathcal{T} \cap \mathcal{S}$ 代表了我们感兴趣也能从零学习的任务，但因为从零学习会消耗监督预算，我们希望从中选择出符合预算的一组从零学习，余下的通过少量数据的迁移学习来实现。我们称 $\mathcal{V} = \mathcal{T} \cup \mathcal{S}$ 为我们的**任务词典** (task dictionary)。最后，我们对视觉任务 t 的定义为一个基于图片的方程 f_t 。

如下图所示，我们收集了一个有四百万张图片的数据题，每张图片均有26个不同视觉任务的标注(ground truth)。这26个任务涵盖了2D的、3D的和语义的任务，构成了本项research的任务词典。因为这26个任务均有标答， $\mathcal{S}$ 也为这26个任务。

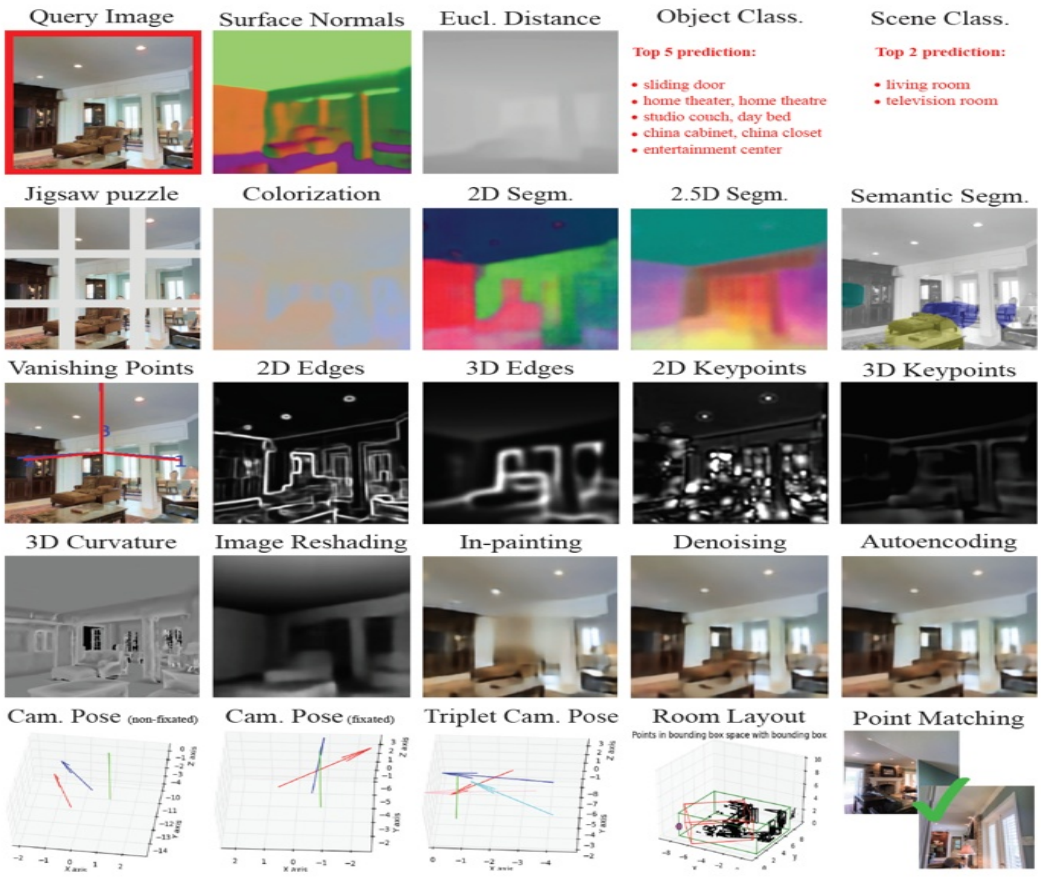


Figure 3: **Task Dictionary.** Outputs of 24 (of 26) task-specific networks for a query (top left). See results of applying frame-wise on a video here.

下面，我们进入第一大阶段，量化视觉任务的关联。

第一步：从零学习

对于每个起始任务，我们为其从零开始学习一个神经网络。为了能更好地控制变量从而比较任务关联，每个任务的神经网络具有相似的 encoder decoder 结构。所有的 encoder 都是相同的类 ResNet 50 结构。因为每个任务的 output 维度各不相同，decoder 的结构对不同的任务各不相同，但都只有几层，远小于 encoder 的大小。（注：CVPR poster session 期间有人问起，decoder 泛指 read out functions，比如 classification 的 FC Layers 也算为 decoder ）

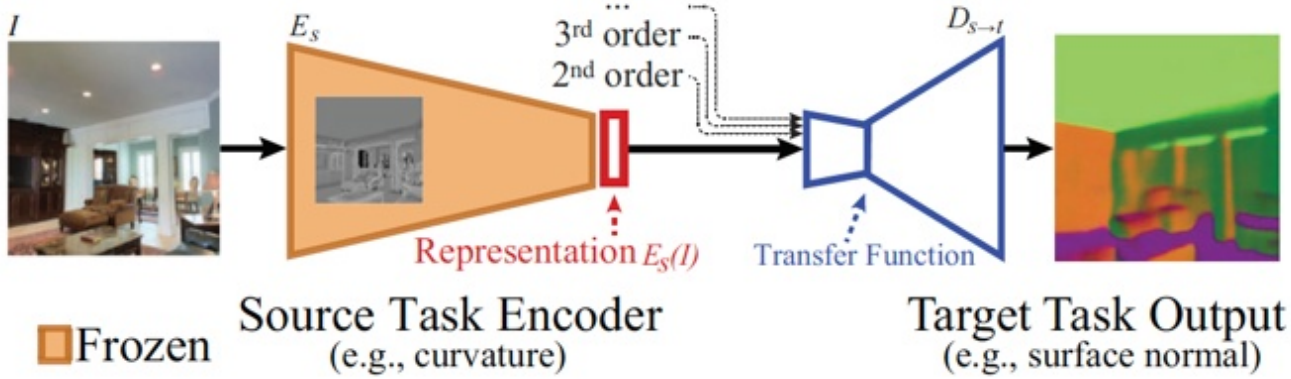


Figure 4: **Transfer Function**. A small readout function is trained to map representations of source task's frozen encoder to target task's labels. If order> 1, transfer function receives representations from multiple sources.

第二步：迁移学习

如上图所示，对于一个起始任务 $s \in \mathcal{S}$ 和一个目标任务 $t \in \mathcal{T}$ ，我们将以 s 的representation作为输入来学习 t 。我们将freeze任务 s 的encoder 参数，并基于encoder的输出(representations) 学习一个浅层神经网络read out function。严谨来讲，如果我们用 E_s 表示 s 的encoder， f_t 表示 t 的标注， L_t 表示 t 的loss函数， $I \in \mathcal{D}$ 来表示图片和迁移训练集， D 表示要迁移学习的浅层神经网络，学习目标为：

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

对于所有 s 和 t 组合，我们均训练了一个 $D_{s \to t}$ 。如下图所示，对于 t ，不同的 $E_s(I)$ 会对 $D_{s \to t}$ 的表现造成不同的影响。更具关联的 s 会为 t 提供更有效的统计信息，从而仅用1/60的训练数据（相较于从零学习）就能取得不错的结果；相反不具备关联的 s 则并不能有此表现。因此，我们认为 $D_{s \to t}$ 在 t 任务中的**表现**可以很好地代表了 s 之于 t 的关联性。

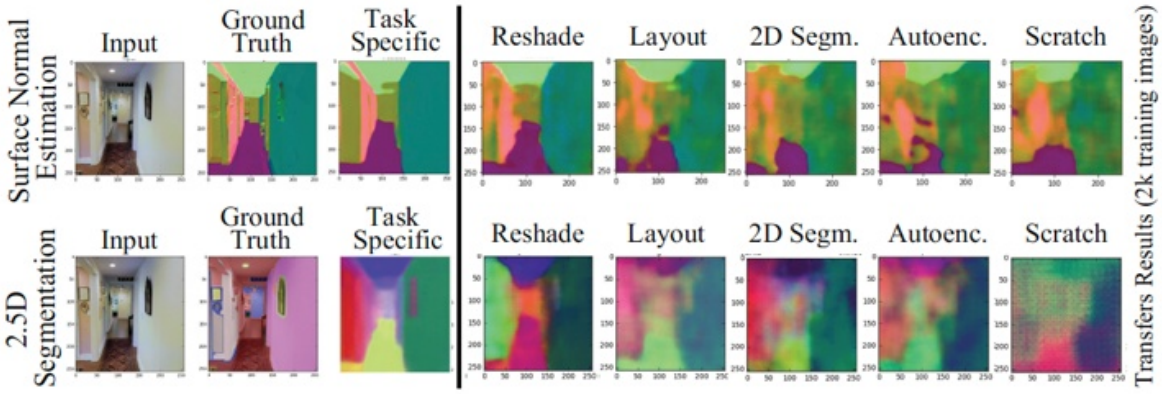


Figure 5: **Transfer results** to normals (upper) and 2.5D Segmentation (lower) from 5 different source tasks. The spread in transferability among different sources is apparent, with reshading among top-performing ones in this case. Task-specific networks were trained on 60x more data. "Scratch" was trained from scratch without transfer learning.

上述迁移代表了任务之间一对一的关联，我们称其为一阶关联。如下图，几个任务之间可能具有互补性，结合几个起始任务的 representations 会对解决目标任务起到帮助。因此，我们也研究了任务之间多对一的关联，我们称其问高阶关联。在这种情况下，我们将几个起始任务的 representation 结合起来当作目标任务的输入，其余细节跟上段类似。

因为高阶的任务组合数量太大，我们基于一阶表现选择了一部分的组合进行迁移学习。对于小于五阶的高阶，我们根据一阶的表现，将前五的所有组合作为输入。对于 n>5 阶，我们选择结合一阶表现前 n 的起始任务作为输入。

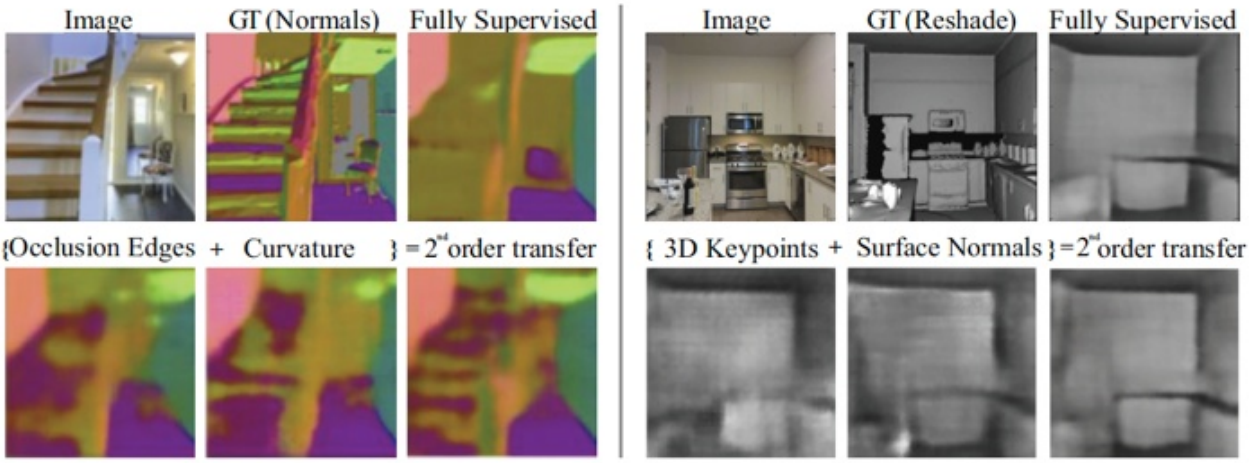


Figure 6: **Higher-Order Transfers.** Representations can contain complementary information. E.g. by transferring simultaneously from 3D Edges and Curvature individual stairs were brought out. See our publicly available interactive transfer visualization page for more examples.

第三步：**Ordinal Normalization(序数归一化)**

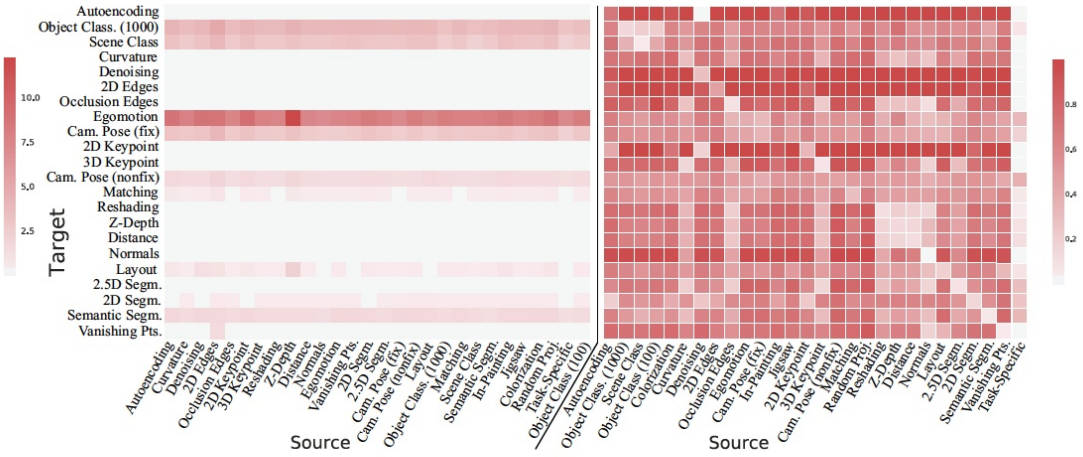


Figure 7: **First-order task affinity matrix** before (left) and after (right) Analytic Hierarchy Process (AHP) normalization. Lower means better transfered. For visualization, we use standard affinity-distance method $dist = e^{-\beta \cdot P}$ (where $\beta = 20$ and e is element-wise matrix exponential).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

这一步的目标为用一个 affinity matrix 量化任务之间的关联。虽然从上步习得的迁移网络中我们获得了许多的loss值 $L_{s \to t}$ ，但因这些loss值来自于不同的loss 函数，它们的值域有很大差别。如果我们把这些loss值直接放入一个矩阵（上图左，纵轴为目标任务、横轴为起始任务），那么这个矩阵内的值及其不均匀，并不能有效反应任务之间的关联。同时，简单的线性规范化也并不能解决问题，因为任务的loss值和表现并不构成线性关系（0.01的 l_2 loss并不代表其表现两倍好于0.02的loss）。由此，我们采用Ordinal Normalization（基于序数的规范化）来将loss值转换为关联度。该方法基于运筹学中的 AHP (Analytic Hierarchy Process)。概括来讲，affinity matrix中的第 (i, j) 个值为利用第 i 个起始任务迁移后，其网络有多大的几率表现好于用第 j 个网络（我们在下文称其为 i 对于 j 的胜率）。

对于每个目标任务 t ，我们构建pairwise tournament矩阵 W_t ，其纵轴和横轴均对应所有的起始任务及我们计算过的高阶组合。给定一个测试集 D_{test} ， W_t 的 (i, j) 项为 s_i 在 D_{test} 的所有图片输入中有多大的几率表现好于 s_j (有几成 $I \in D_{test}$ 会使 $L_t(D_{s_i \to t}(I)) < L_t(D_{s_j \to t}(I))$)。在将 W_t 的值clip到 $[0.001, 0.999]$ ，计算 $W'_t = W_t / W_t^T$ ， W'_t 的 (i, j) 项 $w'_{i,j}$ 现在代表着 s_i 表现好于 s_j 几倍。这样：

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

在把 W'_t 规范化成数值和为1的矩阵后，我们将 s_i 相对于 t 的关联性（抑或可迁移性）定义为 W'_t 的第 i 项principal eigenvector。将所有目标任务的 W'_t 合并起来，我们获得最终的affinity matrix P ，见上图右。

至此第一大阶段完结，我们通过上述 affinity matrix 量化了任务之间的关联性。

第四步：BIP (Binary Integer Programming) 最优化

最后一步，我们要基于 affinity matrix 求得如何最有效地学习一组我们感兴趣的任务。我们可以这个问题想象成一个 subgraph selection 的问题：选择一些任务从零学习，剩下的任务用少量数据进行迁移学习，具体迁移学习的策略由 subgraph 中的 edge 来决定（对一条 directed edge，起始点代表我们从零学习的一个任务，终点代表要进行迁移的目标任务）。基于此，我们可以通过解如下 BIP 最优化问题来得到最优解：

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

这个最优问题有三个限制条件：

1. 如果我们选择了一个迁移，那么迁移的起始任务（可能为高阶起始集）和目标任务均要出现在 subgraph(子图)中；
2. 每个目标任务有且仅有一个迁移（我们将从零学习在途中定义为从自己到自己的迁移，即一条自己到自己的 edge）；
3. 不超过监督预算。

这三个限制条件的具体数学表达如下：

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

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

$$b_i = 0. \tag{5}$$

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

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

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

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

至此，我们通过解最优 subgraph selection 从而获得了最有效迁移学习策略，如下图：

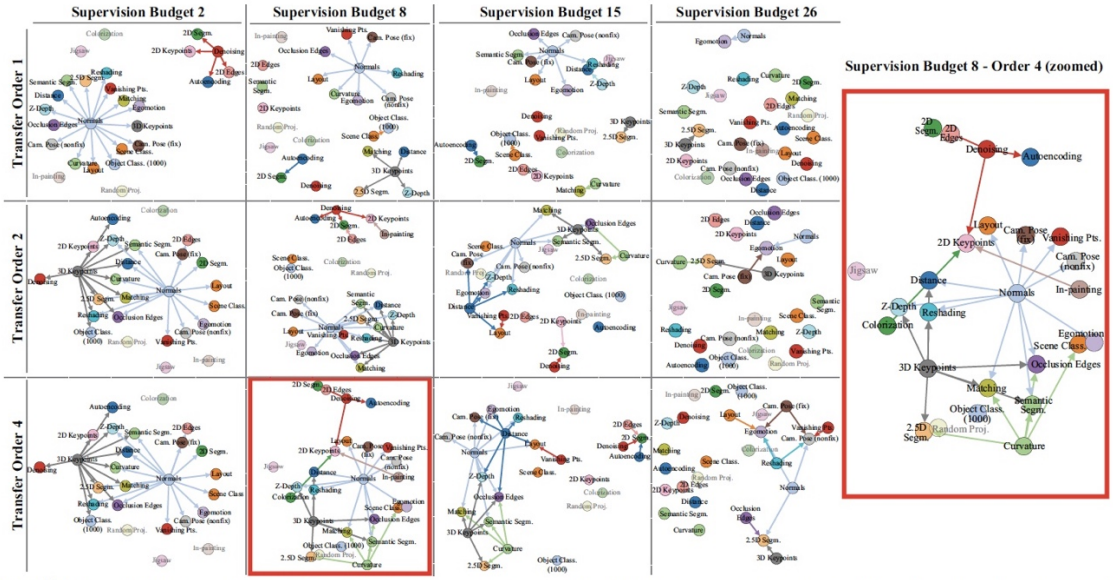


Figure 8: **Computed taxonomies** for solving 22 tasks given various supervision budgets (x-axes), and maximum allowed transfer orders (y-axes). One is magnified for better visibility. Nodes with incoming edges are target tasks, and the number of their incoming edges is the order of their chosen transfer 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.

四、Experiments

Taskonomy 项目训练了 3000+个神经网络，总耗时 ~ 50000 小时的 GPU。从零学习消耗 120k 张图片，迁移学习为 16k 张图片。

我认为现有公众号对 Taskonomy 翻译中最不准确的是对 Taskonomy 实验部分的评论。如文章一开头所说，Taskonomy 的目标为用**有限**的监督预算来**最有效**地解决**一组任务**，并不是将 state of the art 提高百分之几。本文想宣扬的中心思想是计算机视觉届应注重视觉任务间的关联性，并让这些关联性为我们所用。回到本文的具体用途，Taskonomy 的用途有两个：

1. Taskonomy 作为解决**一组**任务的方法。
2. 用 Taskonomy 的任务词典解决一个**只有少量数据的新任务**。

以下试验结果分为两个部分，分别对应以上两点。

一：解决一组任务

如何衡量 Taskonomy 解决一组任务的有效性？我们设定了两个评判标准。

1. 迁移获利 (Gain)：如果我们不进行迁移学习，我们只能基于少量的数据从零学习。迁移获利是指迁移学习相较于从零学习的胜率（见 Ordinal Normalization 部分）。
2. 迁移质量 (Quality)：用少量数据迁移学习相较于用大量数据从零学习的胜率。

下图是 Taskonomy 的迁移获利 (左) 和质量 (右) 的图表。两图的纵轴为所有目标任务，横轴为监督预算，胜率在 0-1 之间。可见，对于一个 26 个任务的目标集，在只有一半甚至 1 / 3 的监督预算时，Taskonomy 计算出的监督分配会使整体表现远远打败从零学习（迁移获利），并近似于（胜率超过 40%）大量数据完全监督学习（迁移质量）。

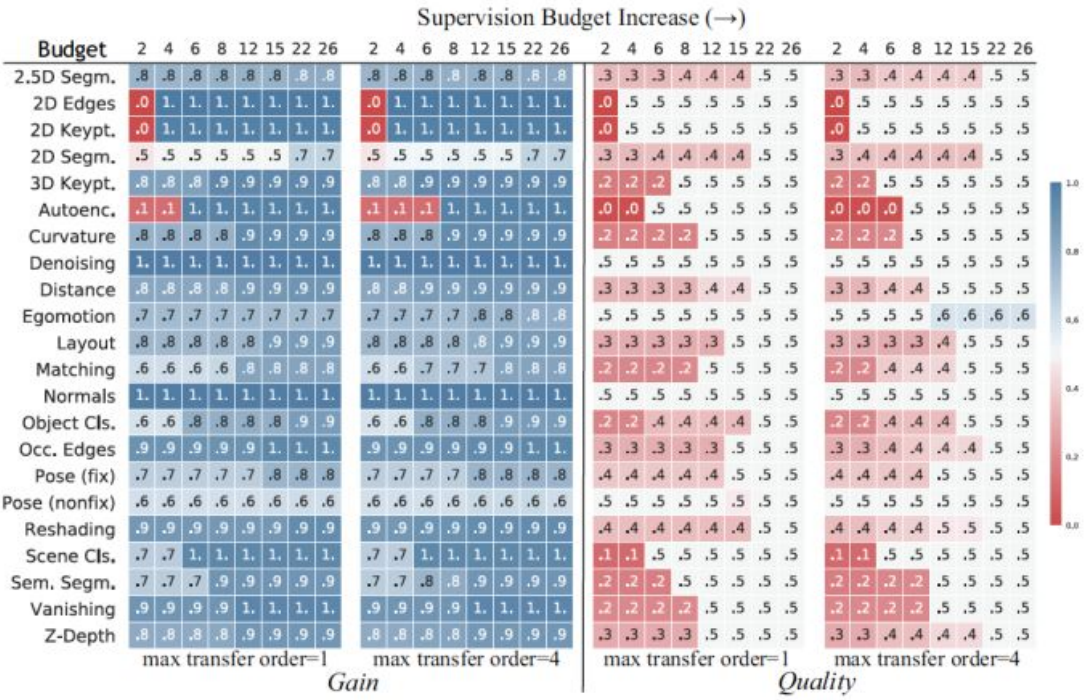
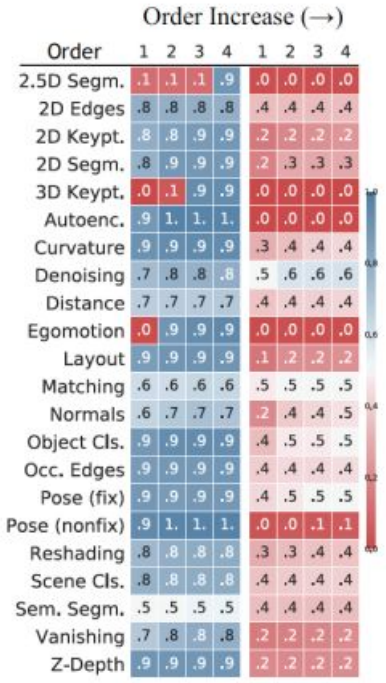


Figure 9: **Evaluation of taxonomy computed for solving the full task dictionary**. Gain (left) and Quality (right) values for each task using the policy suggested by the computed taxonomy, as the supervision budget increases(→). Shown for transfer orders 1 and 4.

二：解决新任务

对于解决新任务，我们可以把我们任务词典里的目标任务当作一个新任务，模拟只有少量数据的情况。实验结果如下，我们可以发现 Taskonomy 的表现超过了现有的行业 pretrained features（包括 imagenet fc7）。



Task	scratch	ImageNet[51]	Wang.[96]	Agrawal.[1]	Zamir.[100]	Zhang.[103]	Noroozi.[68]	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.

五、Conclusion

在 Taskonomy 项目里，我们的目标是着眼于一组任务，并利用任务之间的关联性减少总体数据使用量。为此，我们量化了视觉任务的关联性，并基于求得的 affinity matrix 最优化得到如何分配任务监督数据量。实验表明，视觉任务之间确实存在很强的关联性，我们能通过更少的数据很好地解决一组任务。