Taskonomy: Disentangling Task Transfer Learning

Written by Amir. R Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik and Silvio Savarese

Presented by Peyman Bateni, Lucas Porto, Tanzila Rahman, Vibudh Agrawal

*unless otherwise stated, all figures come from the paper

Timeline for Today

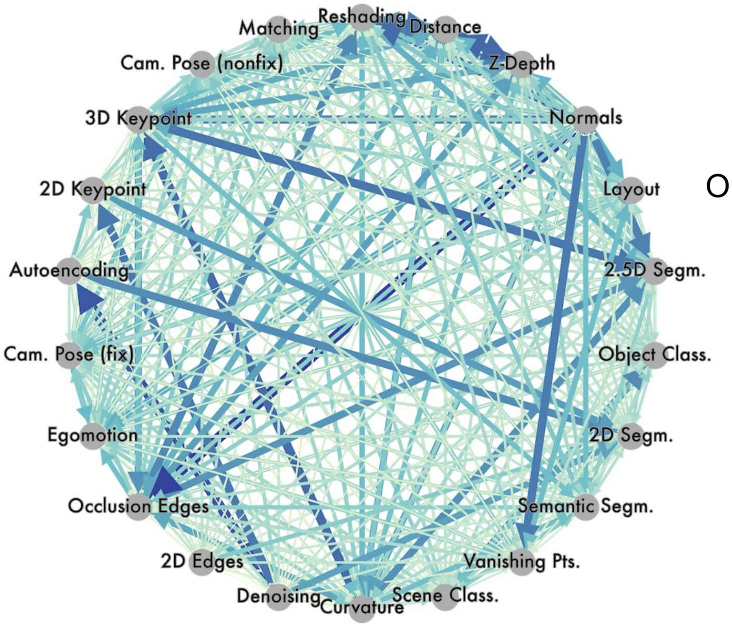
- Introduction
- Related Work
- Method
  - Stage I: Task Specific Modelling
  - Stage II: Transfer Modelling
  - Stage III: Ordinal Normalization using AHP
  - Stage IV: Computing the Global Taxonomy
- Experiments
- Discussion
- Question

Introduction

- **Motivation:** transferability among visual tasks
  - Are visual tasks related? Should we learn a task from scratch every time?
  - Saving time and data: reducing supervision and computation
  - Efficient learning of comprehensive perception models

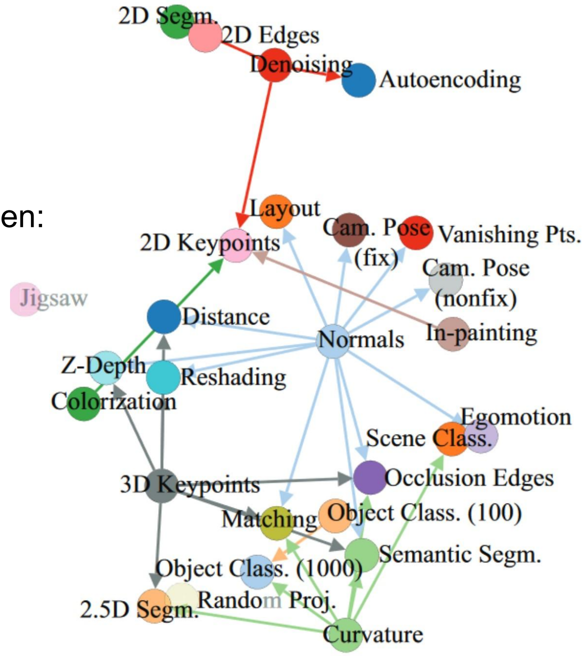
- **Taskonomy (task taxonomy)**: a structure that describes efficient task transfers
  - Directed hypergraph
  - Nodes: tasks
  - Edges: transfers

Taskonomy: Task Taxonomy



Optimize for performance given:

- Supervision budget
- Task importance



Source: https://www.slideshare.net/daynap1204/taskonomy-disentangling-task-transfer-learning-scouty-meetup-2018-feb-14th

Related Work

- **Self-supervised learning:** learning from labels derived directly from the input data
- **Multi-task learning:** learning representations for multiple tasks
- **Domain adaptation:** robustness to different input domains
- **Meta-learning:** similar motivations; higher-level understanding of the learning process

Method - Overview

Task dictionary/bank ($V = \mathcal{T} \cup S$)

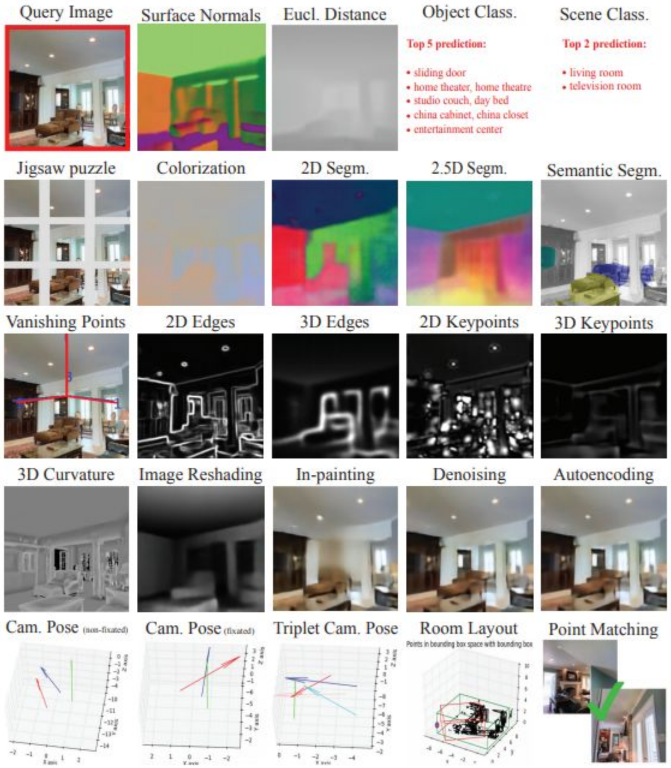
- 26 2D, 2.5D, 3D, and semantic tasks

Dataset

- 4 million images of indoor scenes from about 600 buildings.
- Each image has GT label for all the tasks.

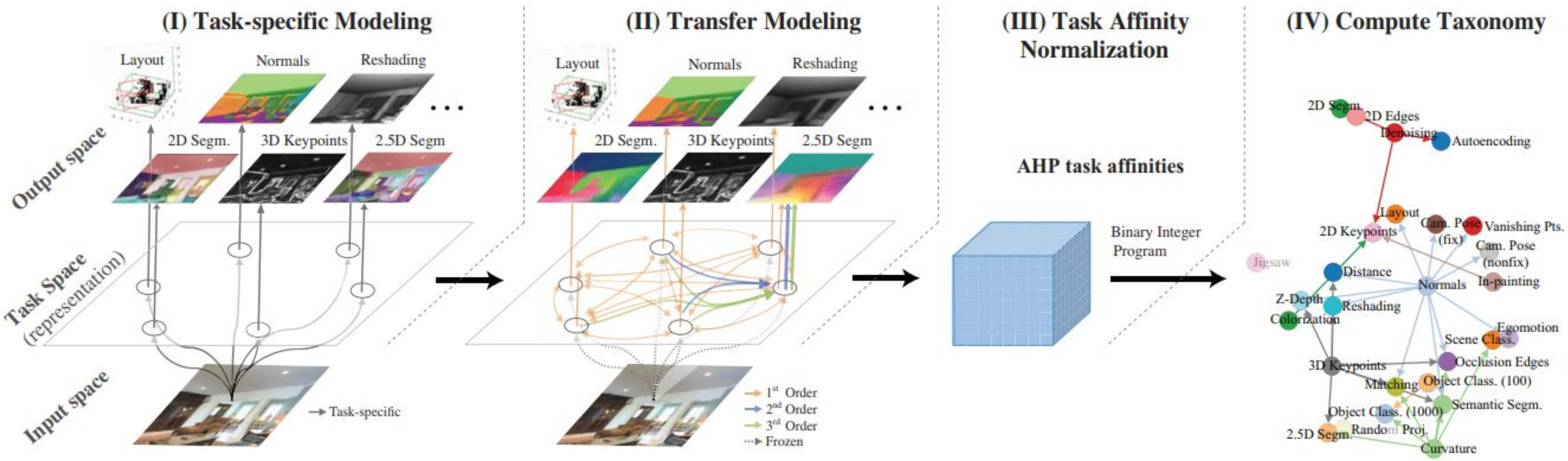
Task specific network

- 26x Network



Task Dictionary

Method - Overview



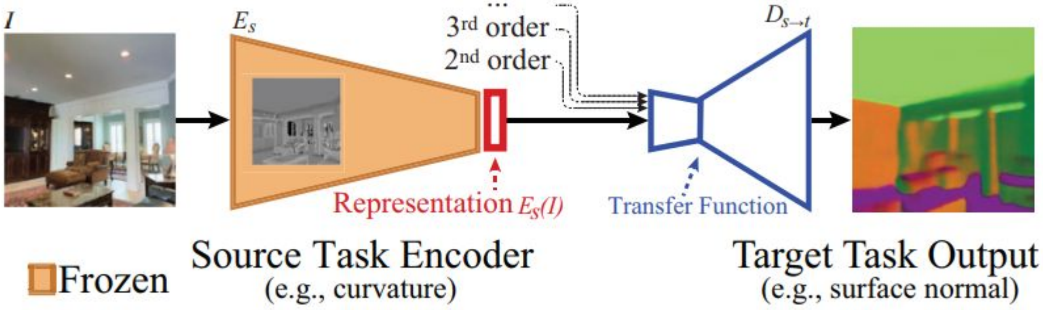
Method - Stage I: Task-Specific Modelling

- Use an encoder-decoder architecture homogeneous across all tasks.
- Encoder: A fully convolutional ResNet-50 without pooling and extract powerful representations.
- Decoder: Different architecture depending on tasks, but smaller than encoder.

Method - Stage II: Task-Specific Modelling

- Transfer network learns a small transfer/readout function ($D_{s\rightarrow t}$).
- $D_{s\rightarrow t}$ is parameterized by $\theta_{s\rightarrow t}$ minimizing the loss L_t:

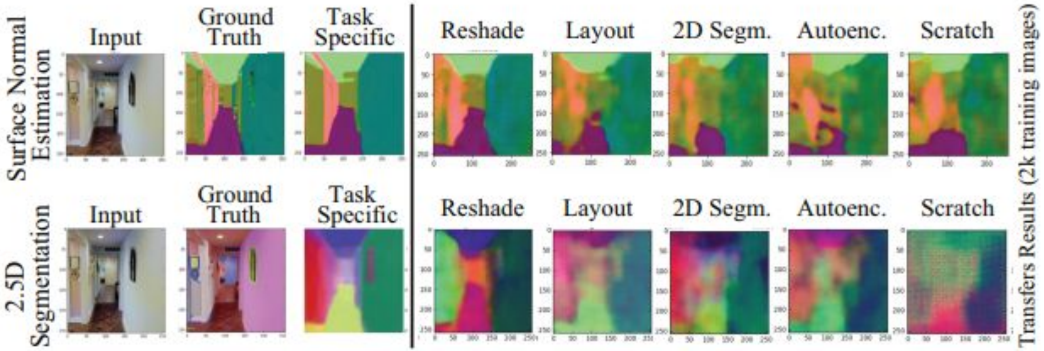
$$D_{s\rightarrow t} := \arg\min_{\theta} \mathbb{E}_{I\in\mathcal{D}}\Big[L_t\Big(D_\theta\big(E_s(I)\big), f_t(I)\Big)\Big]$$



Method - Stage II: Task-Specific Modelling

- Learn a readout function ($D_{s\to t}$).
- $D_{s\to t}$ is parameterized by $\theta_{s\to t}$ minimizing the loss L_t:

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



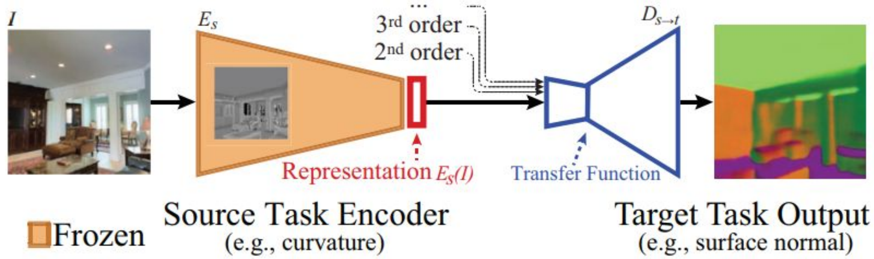
Transfer results

Method - Stage II: Task-Specific Modelling

- Learn a readout function ($D_{s\to t}$).
- $D_{s\to t}$ is parameterized by $\theta_{s\to t}$ minimizing the loss L_t:

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

- Include higher-order transfers to receive multiple representations in the input.



Method - Stage II: Task-Specific Modelling

- Learn a readout function ($D_{s\to t}$).
- $D_{s\to t}$ is parameterized by $\theta_{s\to t}$ minimizing the loss L_t:

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

- Include higher-order transfers to receive multiple representations in the input.

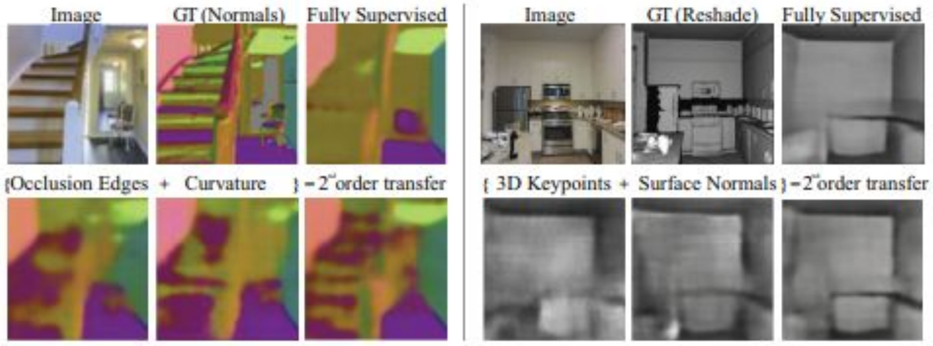
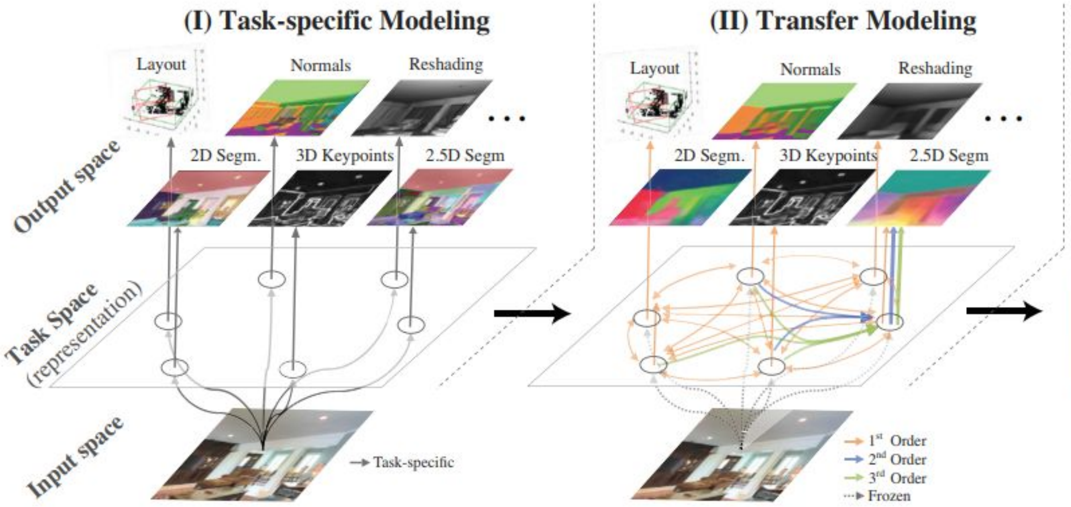
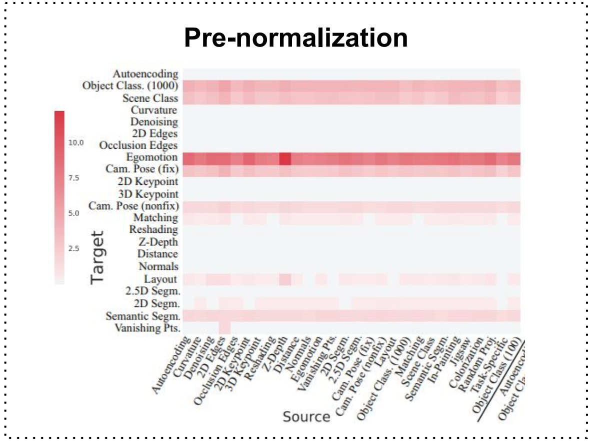


Image
GT (Normals)
Fully Supervised
Image
GT (Reshade)
Fully Supervised
{Occlusion Edges + Curvature }-2nd order transfer
{ 3D Keypoints + Surface Normals}-2nd order transfer

Higher order transfers

Method - Stage II: Task-Specific Modelling





Method - Stage III: Ordinal Normalization using AHP

- A naive normalization can apply to linearly rescale each row of the matrix to the range [0, 1]. But this approach fails when the actual output quality increases at different speeds w.r.t. the loss.
- Apply ordinal normalization derived from Analytic Hierarchy Process (AHP).

Method - Stage III: Ordinal Normalization using AHP

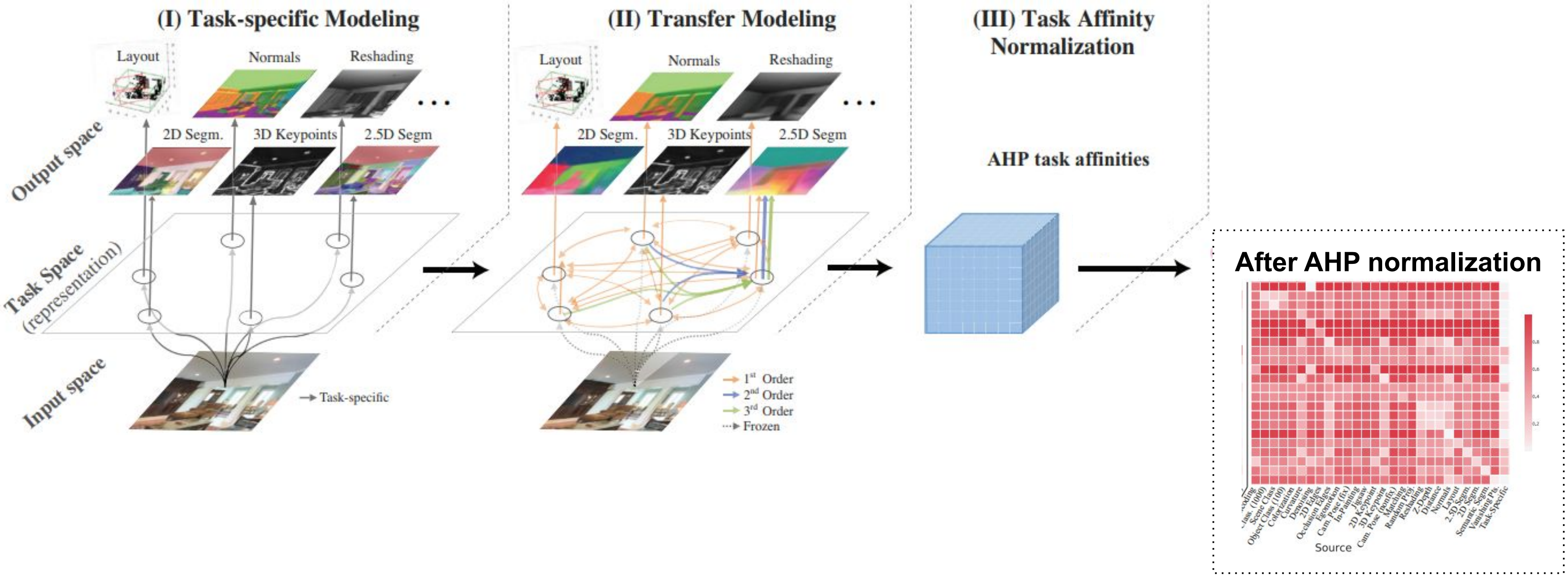
Ordinal Normalization using AHP :

- Construct a pairwise tournament matrix W_t .
- Take win-lose ratio between transfer s_i and s_j.

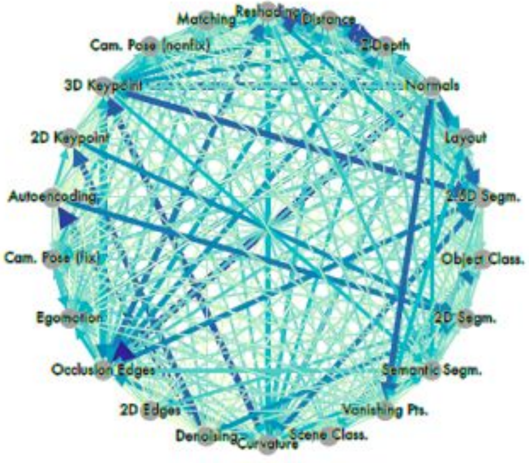
$$w'_{i,j} = \frac{\mathbb{E}_{I \in \mathcal{D}_{test}}[D_{s_i \to t}(I) > D_{s_j \to t}(I)]}{\mathbb{E}_{I \in \mathcal{D}_{test}}[D_{s_i \to t}(I) < D_{s_j \to t}(I)]}$$

- Take the 1st principle component (normalized to sum to 1) of the matrix.
- Create the final matrix by stacking the 1st principle components for all t.

Method - Stage III: Ordinal Normalization using AHP



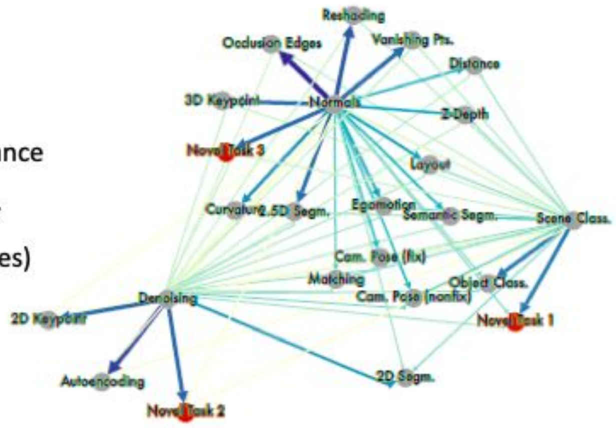
Step IV: Computing the Global Taxonomy (Boolean Integer Programing (BIP))



Post Analytic Hierarchy Process

Maximize collective performance

Minimize supervision (sources)



Post Boolean Integer Programing

Problem formulation

- E (edges in a graph, transfers):

$$\left(\left\{s_1^i, \dots, s_{m_i}^i\right\}, t^i\right)$$

$$\left\{s_1^i, \dots, s_{m_i}^i\right\} \subset S$$

$$t^i \subset T$$

- V (vertices in a graph, tasks):

$$S \cup T$$

- Operations:

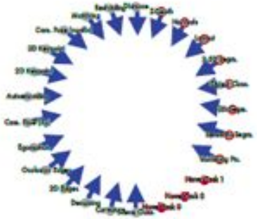
$$\left(\left\{s_1^i, \dots, s_{m_i}^i\right\}, t^i\right) \xrightarrow{\text{sources}} \left\{s_1^i, \dots, s_{m_i}^i\right\}$$

$$\left(\left\{s_1^i, \dots, s_{m_i}^i\right\}, t^i\right) \xrightarrow{\text{target}} t^i$$

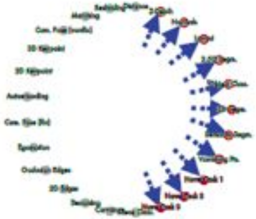
$$\text{Maximize: } \overset{(|E|+|V|,1)}{c}{}^{t}\,\overset{(|E|+|V|,1)}{z}$$

$$\text{Subject to: } \underset{(|E|+2|V|+1,\,|E|+|V|)}{A}x \;\le \underset{(|E|+2|V|+1,1)}{b} \;and\; x \in \{0,1\}^{|E|+|V|}$$

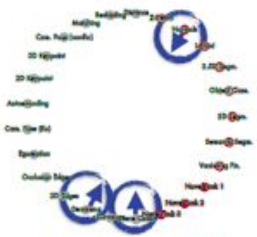
- Constrains:



Constraint I:
only transfer from sources.



Constraint II:
all targets are transferred to.

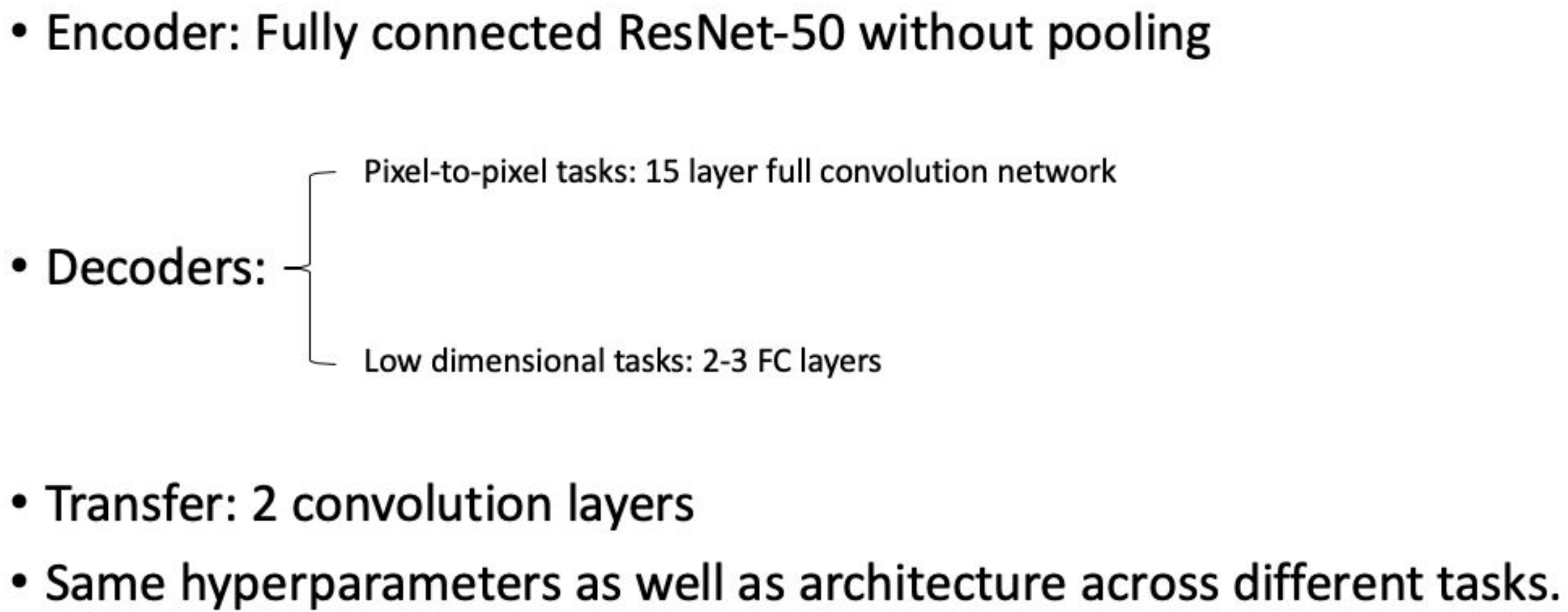


Constraint III:
not exceed budget.

Experiments

- Total tasks in dictionary: 26
- Source only tasks: 4 (colorization, jigsaw puzzle, in-painting, random projection)
- Total transfer functions: ~3000
- GPU hours: 47,886

Network Architecture



- Encoder: Fully connected ResNet-50 without pooling
- Decoders:
  - Pixel-to-pixel tasks: 15 layer full convolution network
  - Low dimensional tasks: 2-3 FC layers
- Transfer: 2 convolution layers
- Same hyperparameters as well as architecture across different tasks.

Data splits

- Dataset: 4 million images
- Training: 120k
- Validation: 16k
- Test: 17k
- Task specific network (Training dataset)
- Transfer network (validation dataset, ranging from 1k – 16K)

Experiments I - Evaluation of Computed Taxonomies

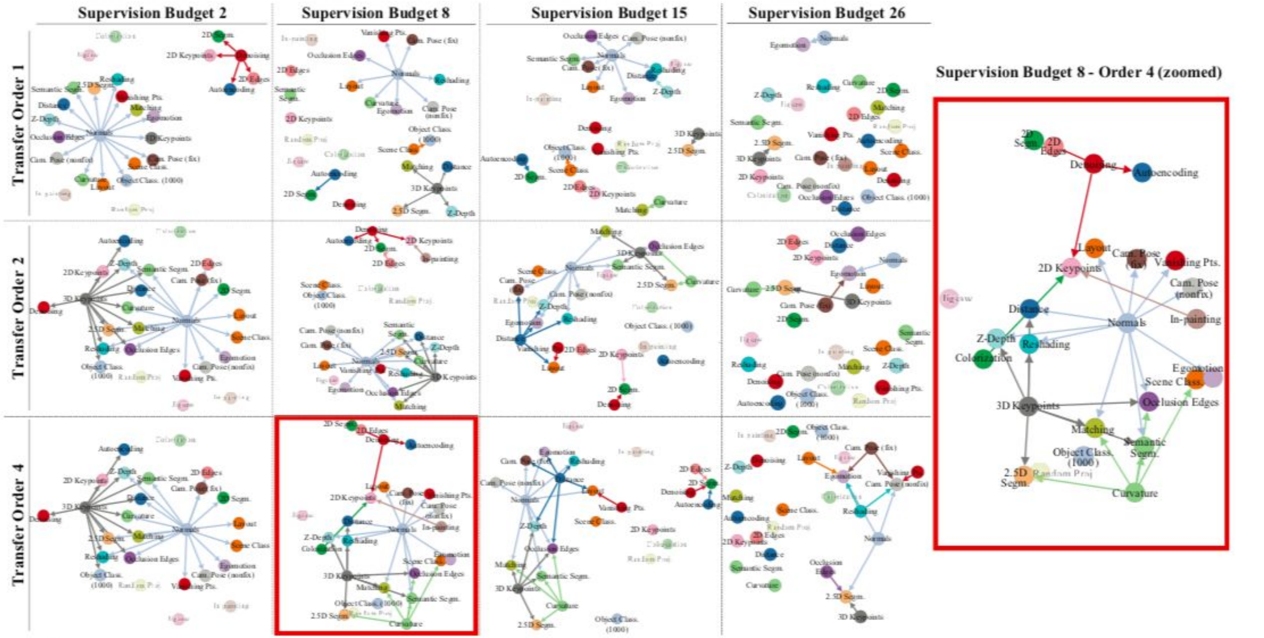


Figure 8: **Computed taxonomies** for solving 22 tasks given various supervision budgets (x-axes), and maximum allowed transfer orders (y-axes). One is magnified for better visibility. Nodes with incoming edges are target tasks, and the number of their incoming edges is the order of their chosen transfer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

Experiments I - Evaluation of Computed Taxonomies

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).

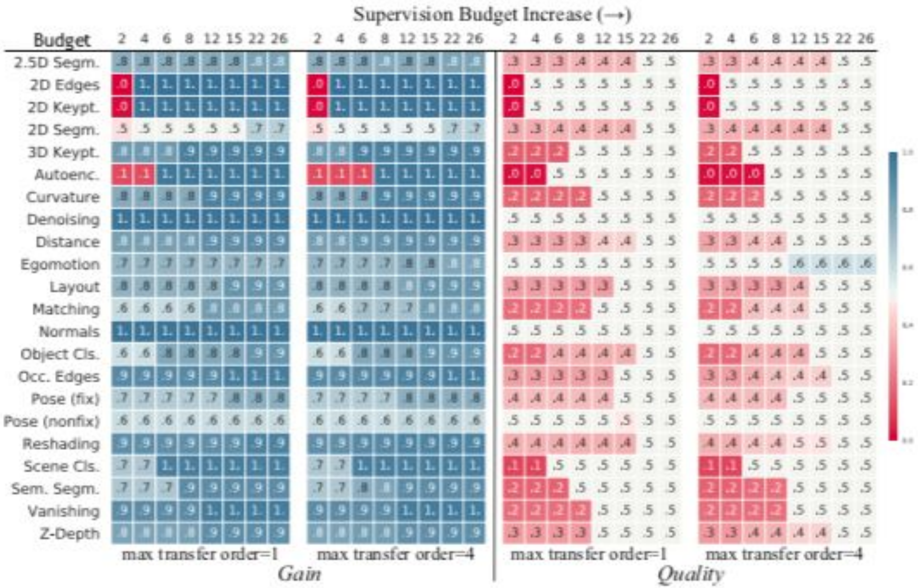
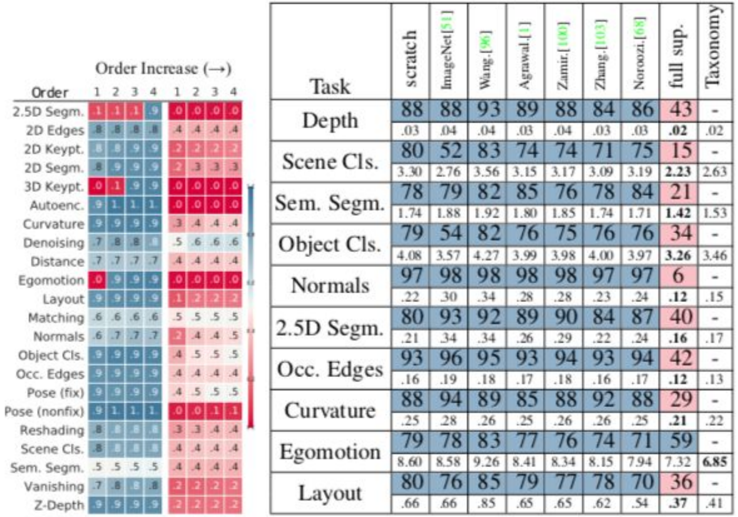


Figure 9: **Evaluation of taxonomy computed for solving the full task dictionary**. Gain (left) and Quality (right) values for each task using the policy suggested by the computed taxonomy, as the supervision budget increases(→). Shown for transfer orders 1 and 4.

Experiments II - Generalization to Novel Tasks

- **Carefully transferring policies depending on the target is superior to fixed transfers**
- **Although taxonomy transfer policies lose to fully supervised networks, in most cases results get close to the gold standard with win rate at 40%**



Order Increase (→)

Order	1	2	3	4	1	2	3	4
2.5D Segm.	.1	.1	.1	.9	.0	.0	.0	.0
2D Edges	.8	.8	.8	.8	.4	.4	.4	.4
2D Keypt.	.8	.8	.9	.9	.2	.2	.2	.2
2D Segm.	.8	.9	.9	.9	.2	.3	.3	.3
3D Keypt.	.0	.1	.9	.9	.0	.0	.0	.0
Autoenc.	.9	1.	1.	1.	.0	.0	.0	.0
Curvature	.9	.9	.9	.9	.3	.4	.4	.4
Denoising	.7	.8	.8	.8	.5	.6	.6	.6
Distance	.7	.7	.7	.7	.4	.4	.4	.4
Egomotion	.0	.9	.9	.9	.0	.0	.0	.0
Layout	.9	.9	.9	.9	.1	.2	.2	.2
Matching	.6	.6	.6	.6	.5	.5	.5	.5
Normals	.6	.7	.7	.7	.2	.4	.4	.5
Object Cls.	.9	.9	.9	.9	.4	.5	.5	.5
Occ. Edges	.9	.9	.9	.9	.4	.4	.4	.4
Pose (fix)	.9	.9	.9	.9	.4	.5	.5	.5
Pose (nonfix)	.9	1.	1.	1.	.0	.0	.1	.1
Reshading	.8	.8	.8	.8	.3	.3	.4	.4
Scene Cls.	.8	.8	.8	.8	.4	.4	.4	.4
Sem. Segm.	.5	.5	.5	.5	.4	.4	.4	.4
Vanishing	.7	.8	.8	.8	.2	.2	.2	.2
Z-Depth	.9	.9	.9	.9	.2	.2	.2	.2

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.

Experiments III - Significance Test of the Structure

- **Outperforms all other randomly assigned connective networks, indicating the significant and existence of an underlying connective structure between the tasks**

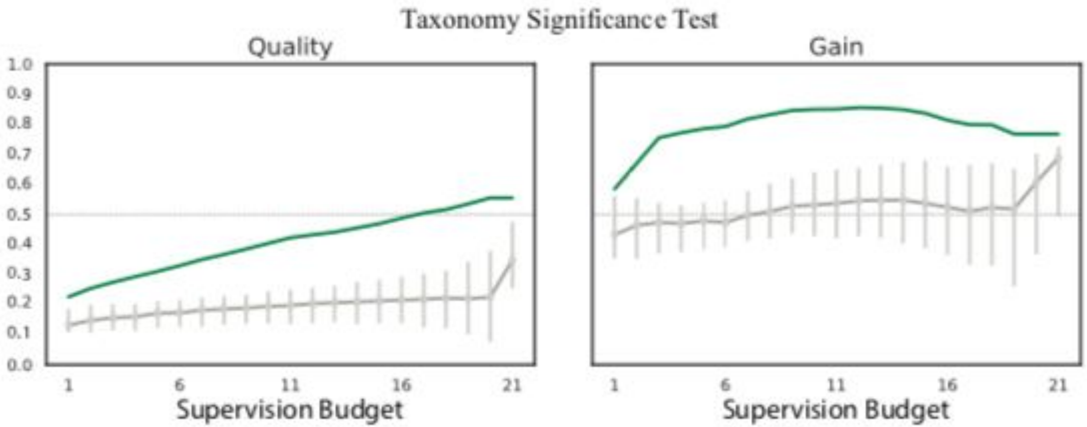


Figure 11: **Structure Significance.** Our taxonomy compared with random transfer policies (random feasible taxonomies that use the maximum allowable supervision budget). Y-axis shows *Quality* or *Gain*, and X-axis is the supervision budget. Green and gray represent our taxonomy and random connectivities,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

Experiments IV - Evaluation on MIT Places & ImageNet

- Spearman's rho between taxonomy ranking and the Top-1 ranking is **0.857 on Places and 0.823 on ImageNet** showing a notable correlation

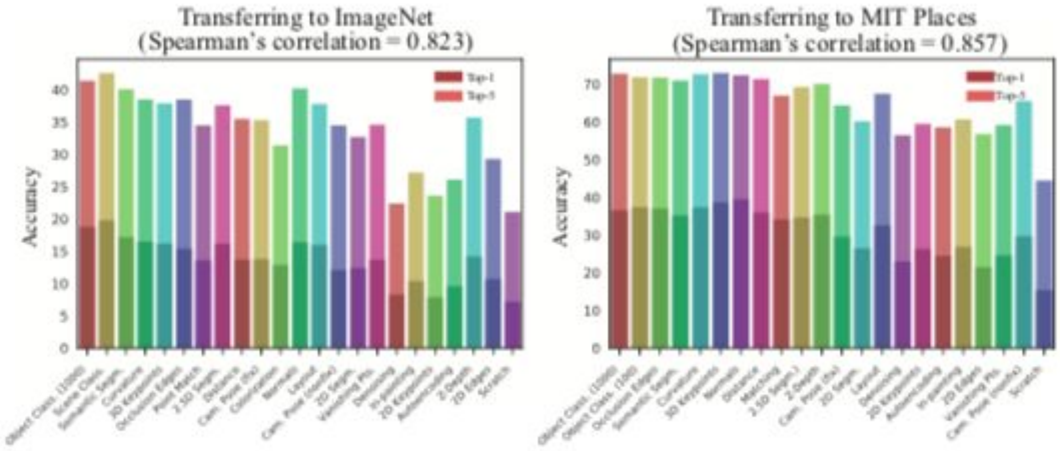


Figure 12: **Evaluating the discovered structure on other datasets: ImageNet [78] (left) for object classification and MIT Places [104] (right) for scene classification.** Y-axis shows accuracy on the external benchmark while bars on x-axis are ordered by taxonomy's predicted performance based on our dataset. A monotonically decreasing plot corresponds to preserving identical orders and perfect generalization.

Experiments V - Universality of the Structure

System choices

I. **Architecture of Task-Specific Networks**
II. **Architecture of Transfer Function Networks**
III. **Amount of Data Available for Training Networks**
IV. **Datasets**
V. **Data Splits**
VI. **Choice of Dictionary**

Remarkably stable leading to almost no change in the output taxonomy

Experiments VI - Task Similarity Tree

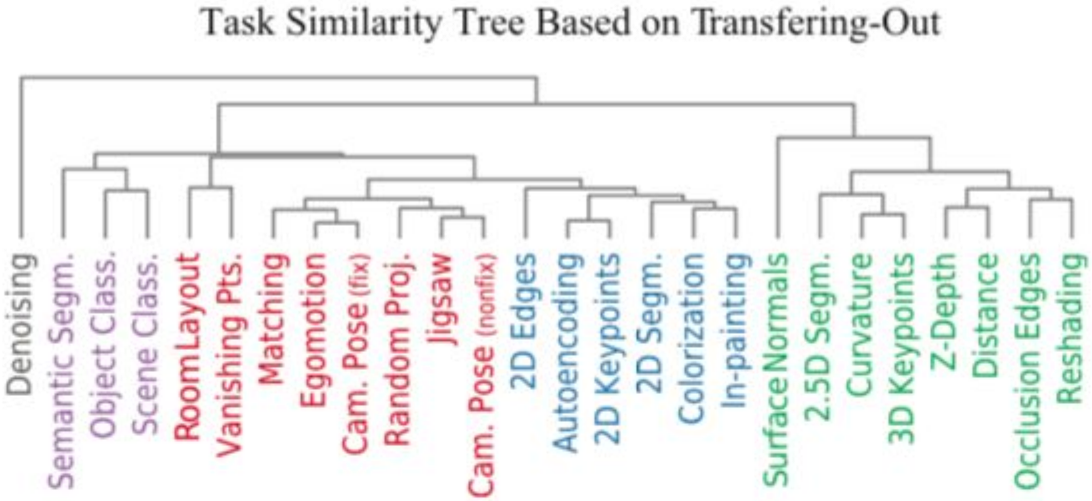


Figure 13: **Task Similarity Tree.** Agglomerative clustering of tasks based on their transferring-out patterns (i.e. using columns of normalized affinity matrix as task features). 3D, 2D, low dimensional geometric, and semantic tasks clustered together using a fully computational approach.

Limitations and Discussion

- **Model and data specific**
  - Taxonomy is computed w.r.t. a particular model and dataset
- **Compositionality**
  - Tasks as composition of subtasks
- **Non-visual tasks**
  - Does this method apply to non-visual tasks?
- **Lifelong/Continuous learning**
  - Expanding the taxonomy after computing it

Thank You

- **We’re happy to answer any questions :)**