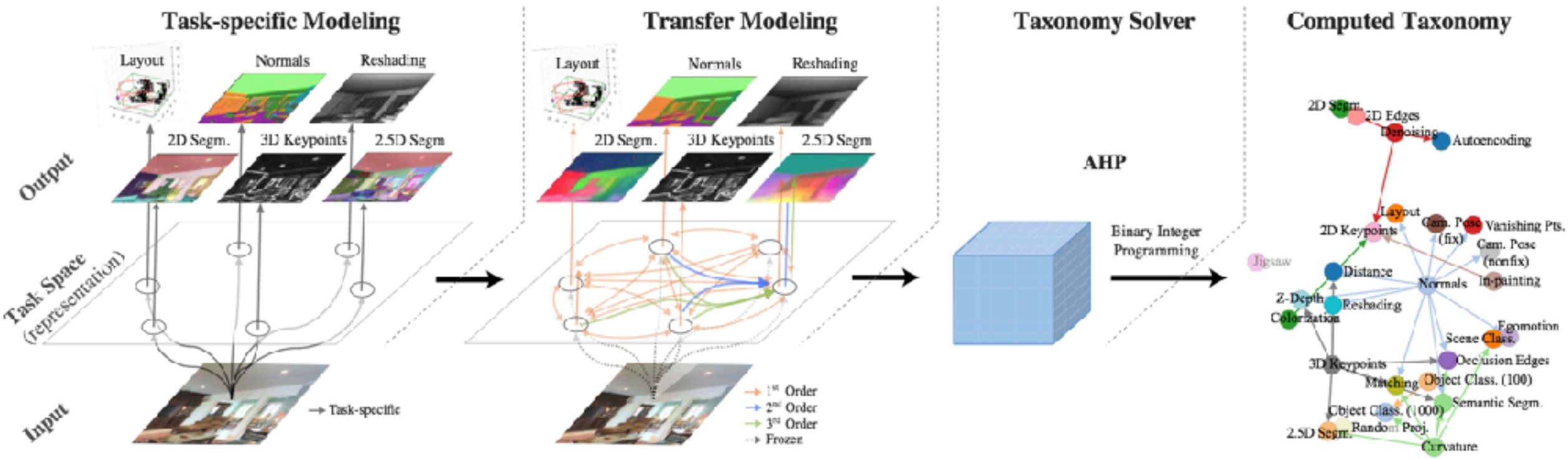
Taskonomy: Disentangling Task Transfer Learning

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/

focal Systems



Process overview. The steps involved in creating the taxonomy.

focal Systems

Learning Tasks

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) [68, 72, 17, 103, 100, 69]. 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.

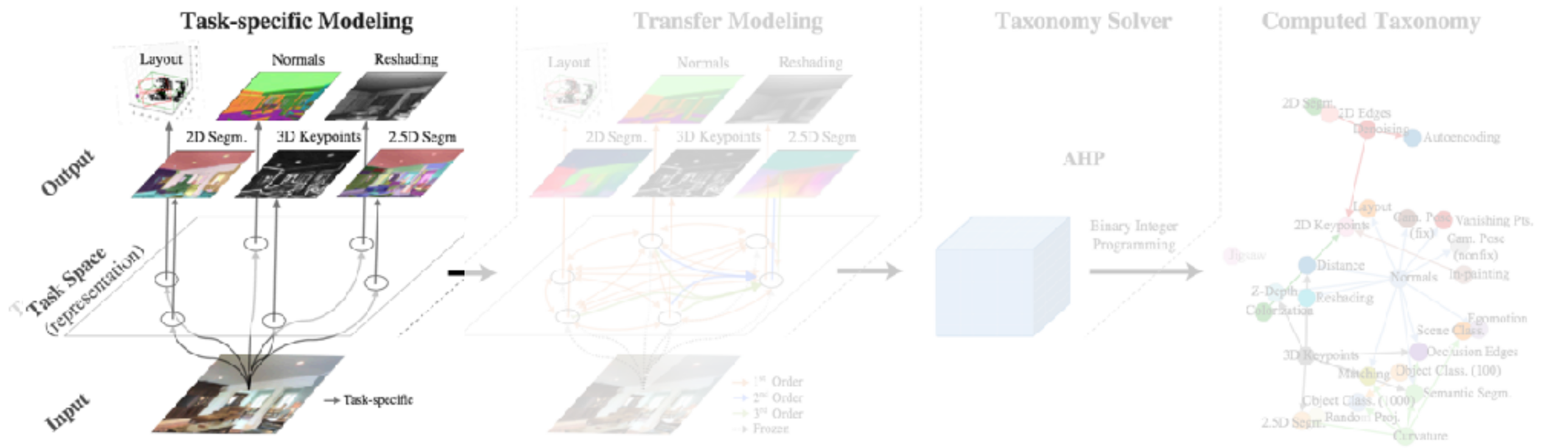
Unsupervised learning is concerned with the redundancies in the input domain and leveraging them for forming compact representations, which are usually agnostic to the downstream task [8, 49, 20, 9, 32, 77]. Our approach is not unsupervised by definition as it is not agnostic to the tasks. Instead, it 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 [21, 31, 28], optimization [2, 82, 48], or certain architectural mechanisms [27, 30, 87, 65]. 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 [50, 18]. Multi-task learning has experienced recent progress and the reported advantages are another support for existence of a useful structure among tasks [93, 100, 50, 76, 73, 50, 18, 97, 61, 11, 66]. 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 [44, 99, 5, 80, 52, 26, 36]. It often addresses a shift in the *input* domain, e.g. webcam images to D-SLR [47], 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.

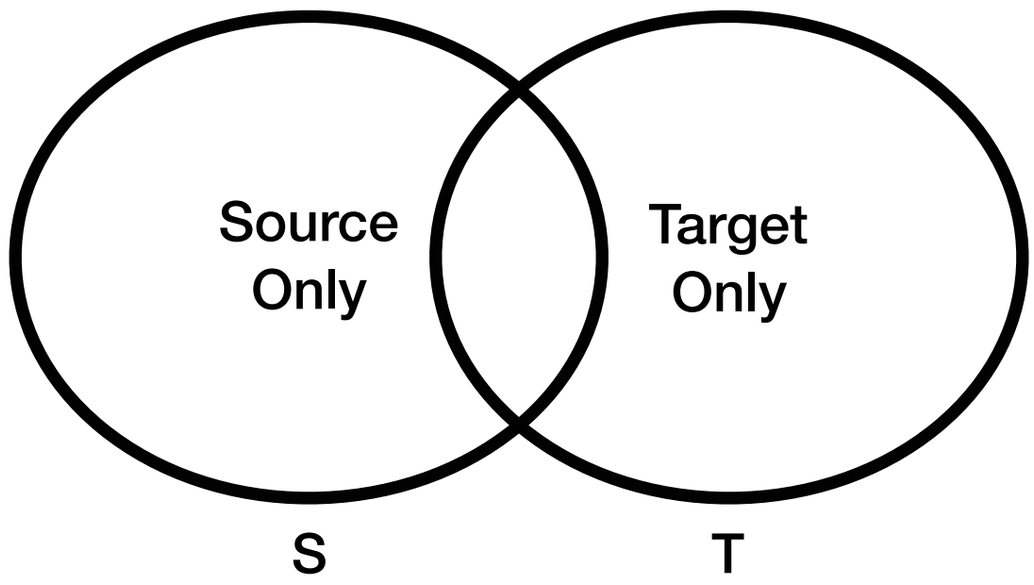
focal Systems



rocess overview. The steps involved in creating the taxonomy.

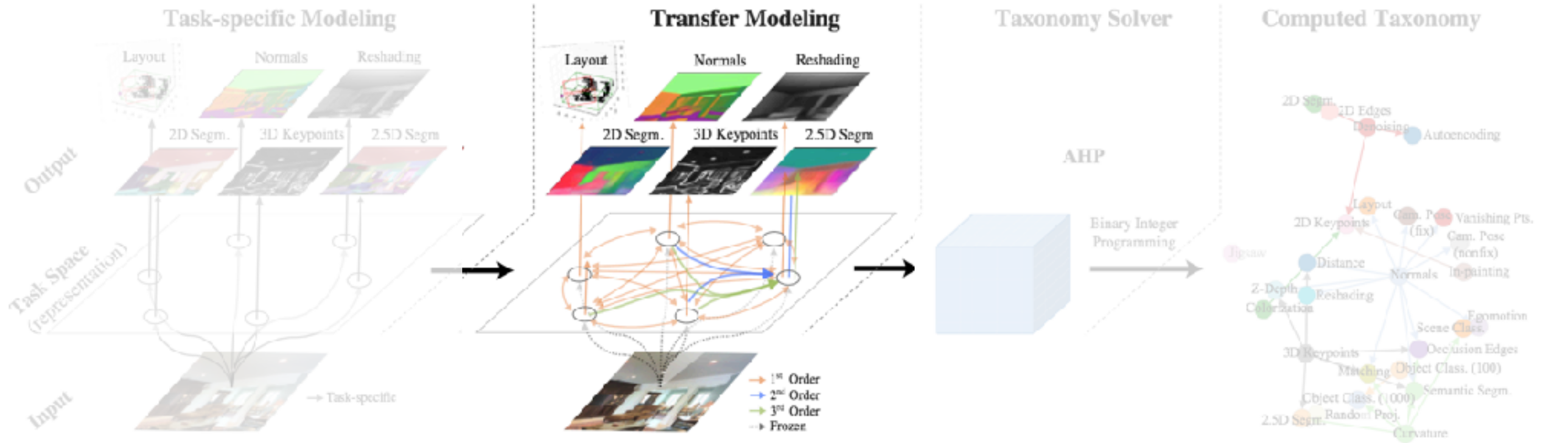
max(Performance on task 1 + Performance on task 2 …)
s.t. supervision budget (# of source tasks)

V=S + T



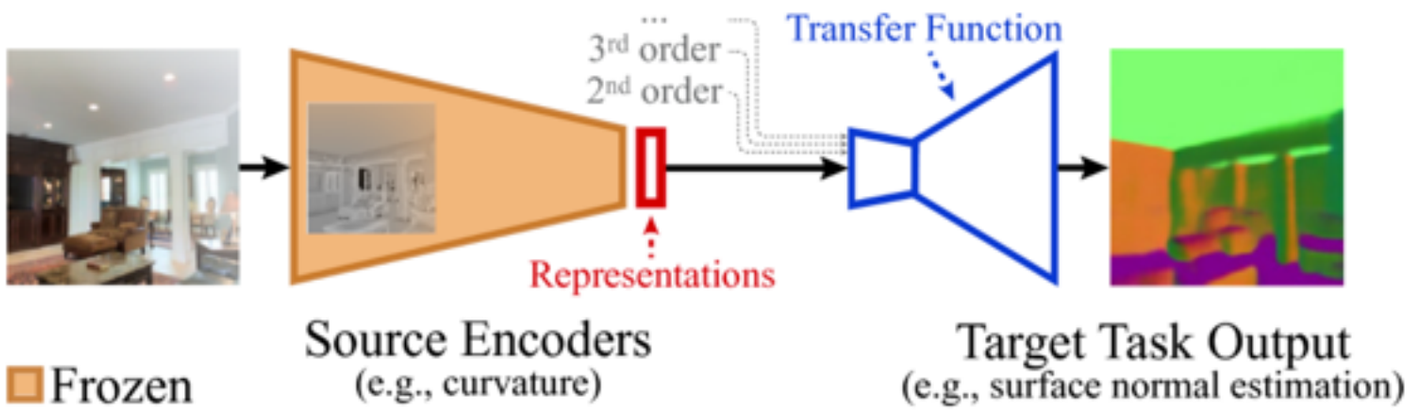
1. 4m images. 120k train, 16k val, 17k test
2. 26 tasks maximum
3. Used resent50
4. Transfer models were 2 layer convs
5. 22x25 transfer funcs (first order)
6. Took 5.5 years of GPU time! :(

focal Systems



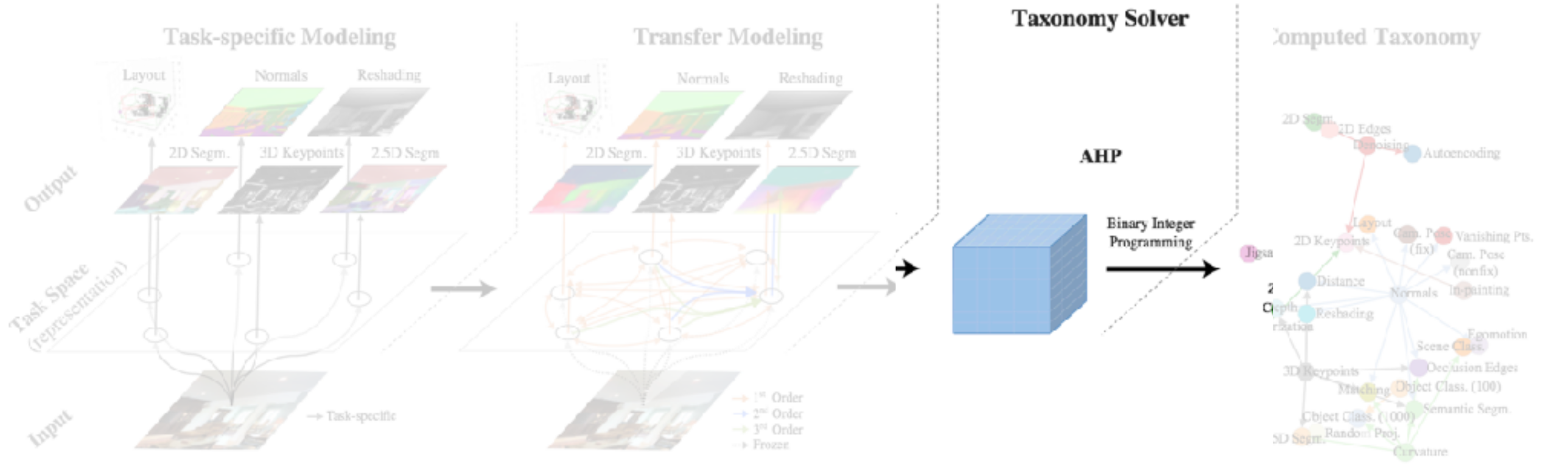
Process overview. The steps involved in creating the taxonomy.

Computer all feasible transfers between sources and target, for all targets. (higher order is allowed)



Transfer Function. We use one or multiple source tasks to predict a target task's output.

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



Process overview. The steps involved in creating the taxonomy.

Normalize the Task Affinity Matrix:

1. **Typical normalization (linearize to 0-1) will not work since different losses!**
2. **Compute win ratios**
3. **Compute weight matrix**
4. **Normalize via Analytical Hierarchy Process**

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

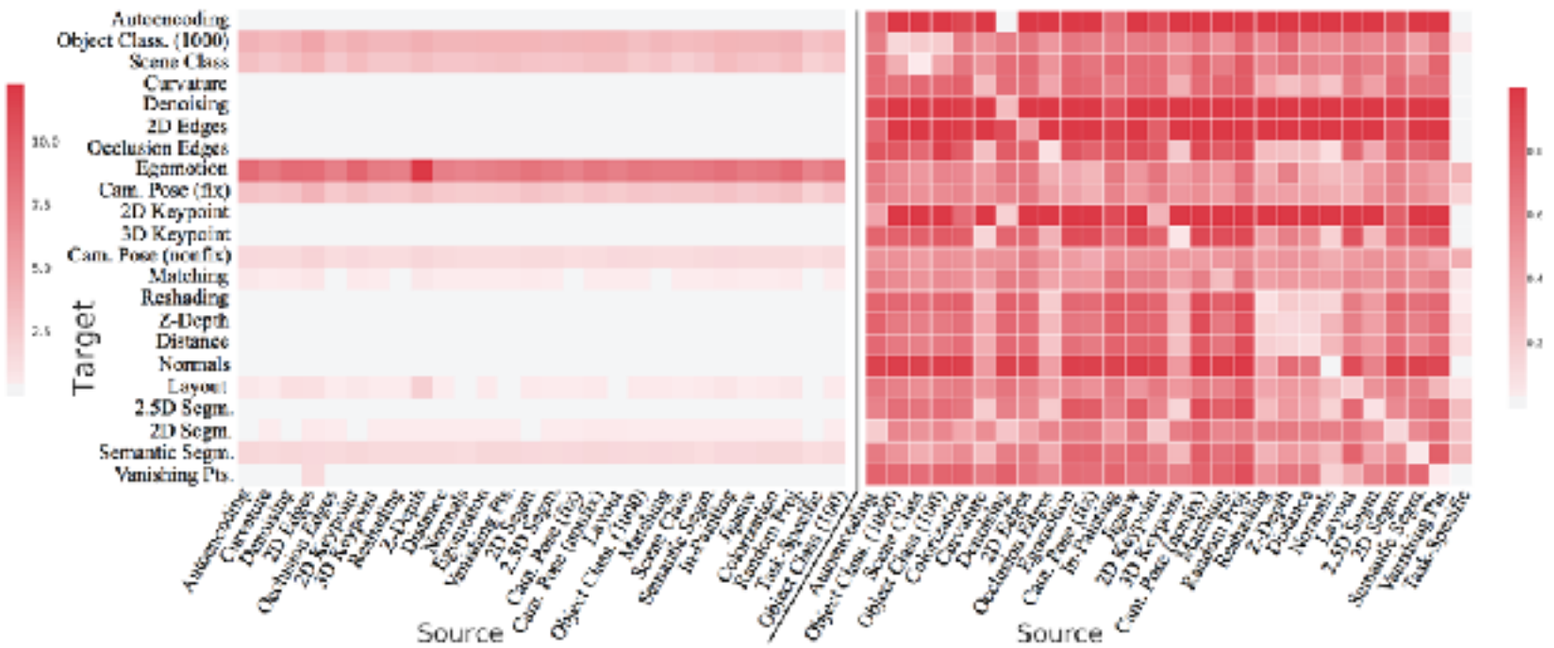
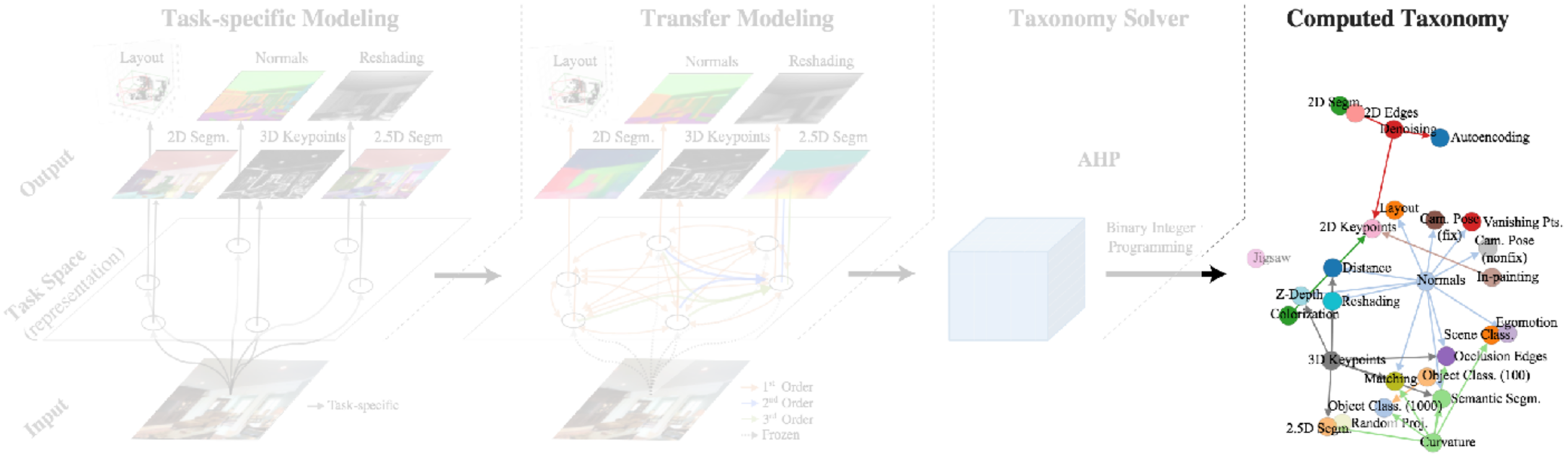


Figure 7: **First-order task affinity matrix** before (left) and after (right)

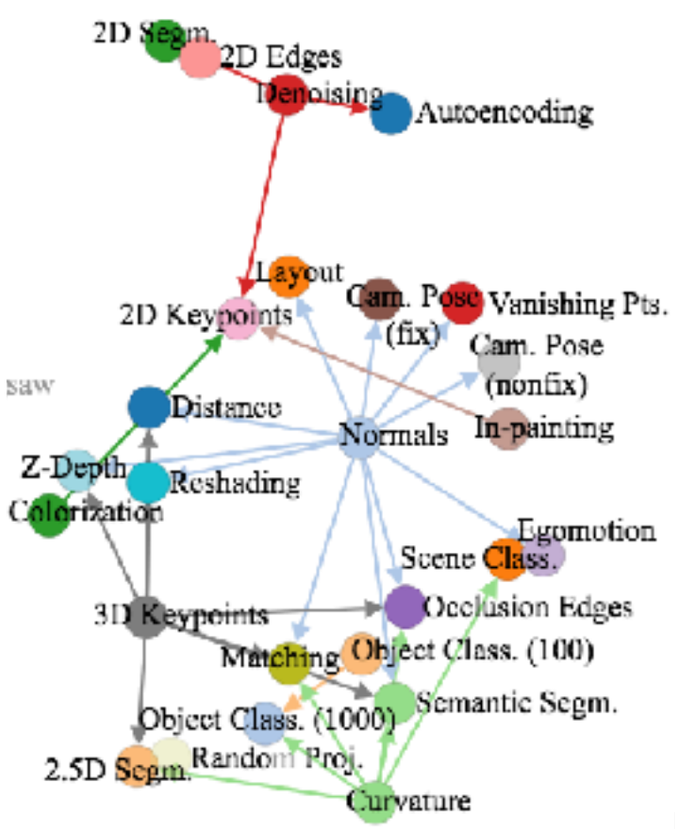


Process overview. The steps involved in creating the taxonomy.

Given a new target task, synthesize the hypergraph and find the optimal transfer policy:

1. **For all source models, freeze submodel, train transfer model, compute performance**
2. **Find the optimal subgraph (tasks are nodes and transfers are edges) and pick the ideal source nodes and the best edges using Boolean Integer Programming**
3. **Use the subselected models and transfer models to forward pass**

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



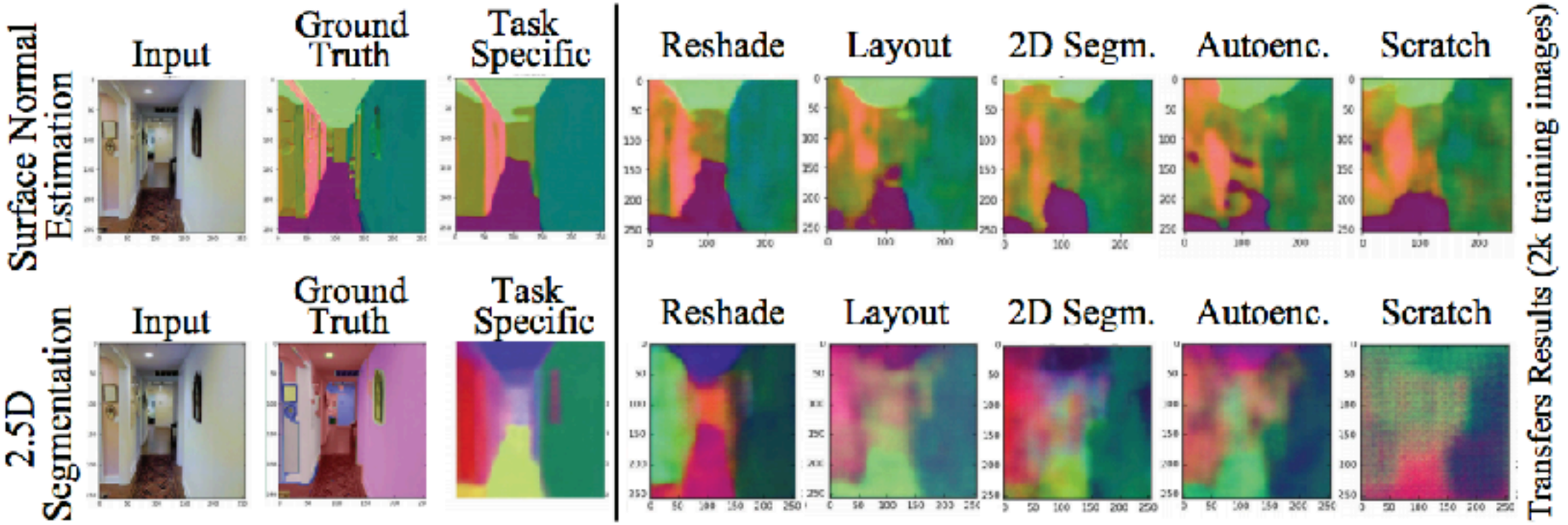


Figure 5: **Transfer results** to normals (upper) and 2.5D Segmentation (lower) from 5 different source tasks. The spread in transferability among different sources is apparent, with reshading among top-performing ones in this case. Task-specific networks were trained on 60x more data. "Scratch" was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

focal Systems

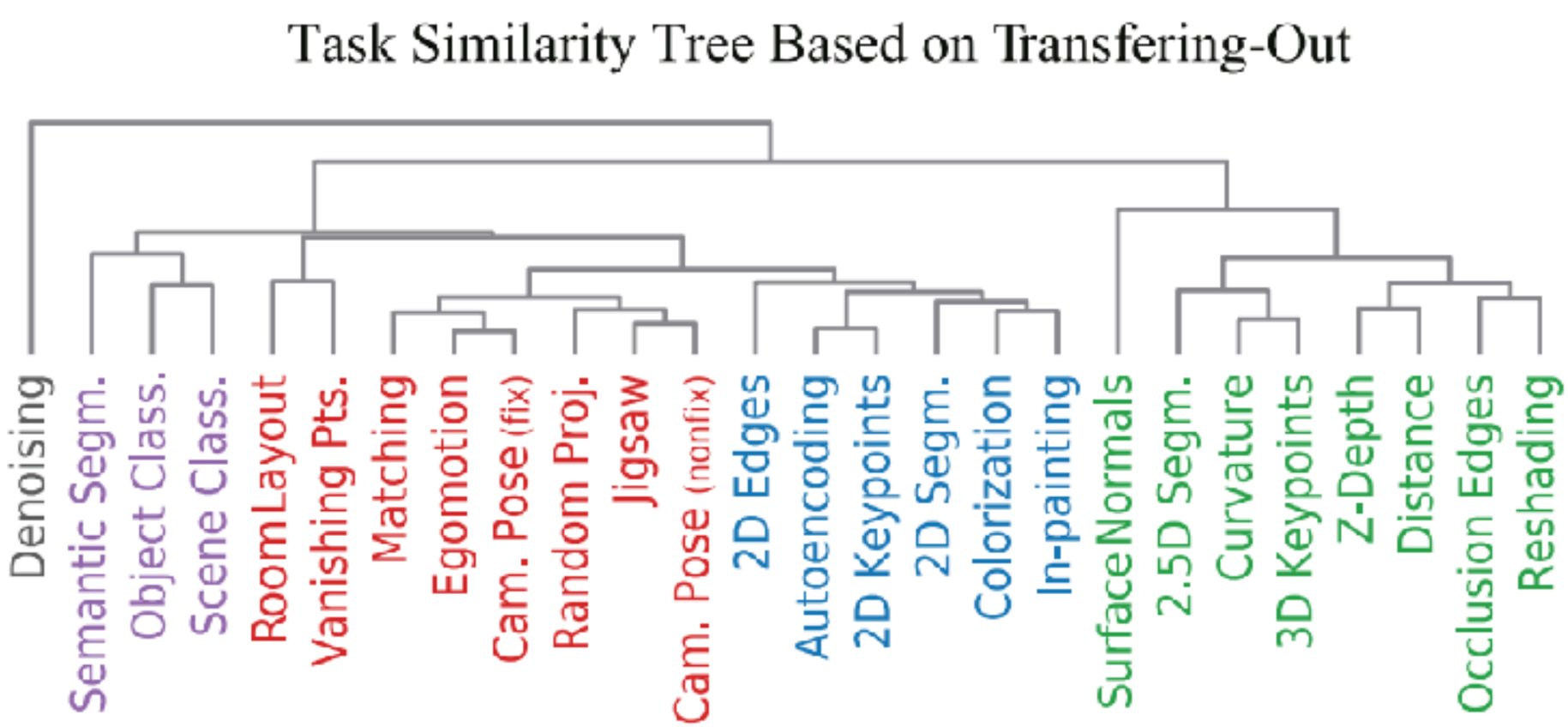


Figure 13: **Task Similarity Tree.** Agglomerative clustering of tasks based on their transferring-out patterns (i.e. using columns of normalized affinity matrix as task features). 3D, 2D, low dimensional geometric, and semantic tasks clustered together using a fully computational approach.

focal
Systems

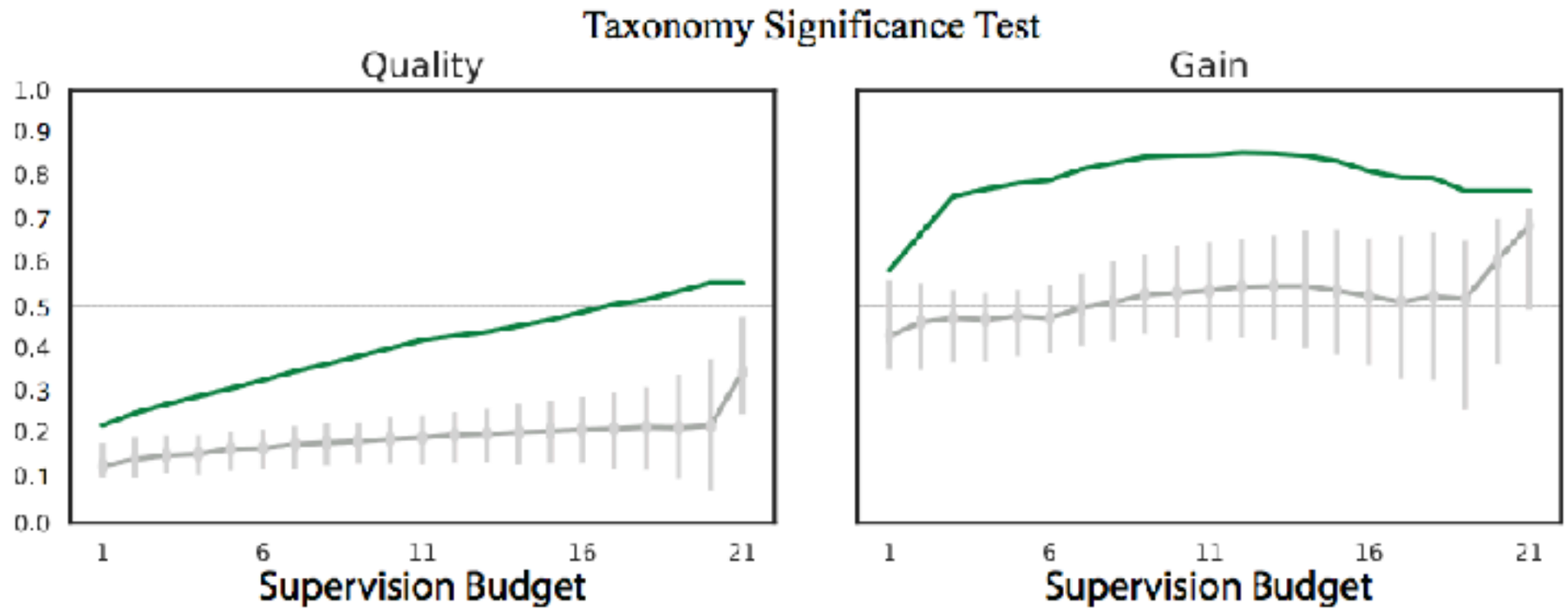
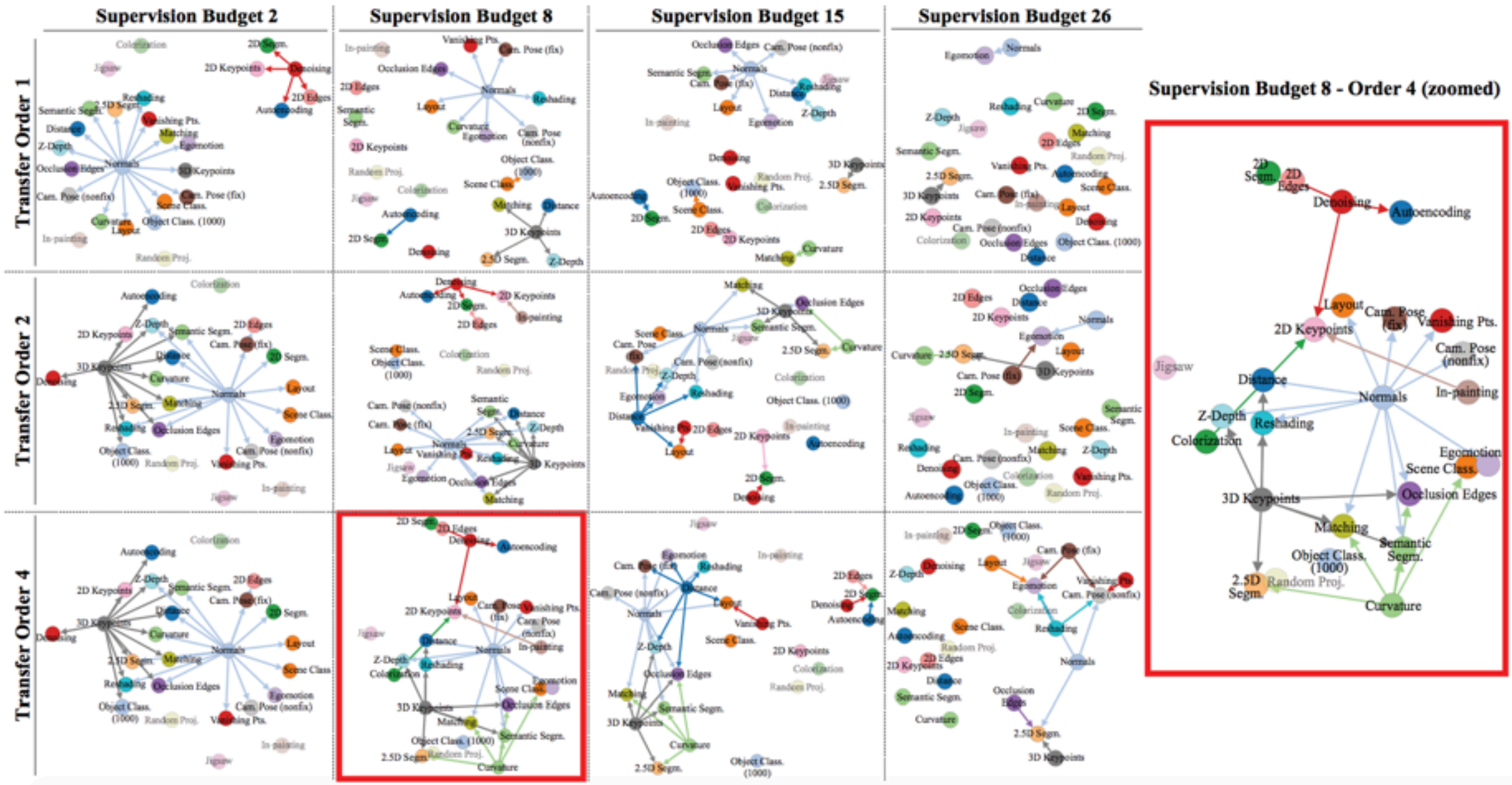


Figure 11: **Structure Significance.** Our taxonomy compared with random transfer policies (random feasible taxonomies that use the maximum allowable supervision budget). Y-axis shows *Quality* or *Gain*, and X-axis is the supervision budget. Green and gray represent our taxonomy and random connectivities,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

focal
Systems


Supervision Budget 2
Supervision Budget 8
Supervision Budget 15
Supervision Budget 26
Transfer Order 1
Transfer Order 2
Transfer Order 4
Supervision Budget 8 - Order 4 (zoomed)
2D Segm.
2D Edges
Denoising
Autoencoding
Layout
2D Keypoints
Cam. Pose (fix)
Vanishing Pts.
Cam. Pose (nonfix)
In-painting
Distance
Normals
Z-Depth
Reshading
Colorization
Egomotion
Scene Class.
Occlusion Edges
3D Keypoints
Matching
Semantic Segm.
2.5D Segm.
Curvature
Object Class. (1000)
Jigsaw
Random Proj.

LIVE DEMO!

focal
Systems