Taskonomy:
Disentangling Task Transfer Learning

Contents

Contents

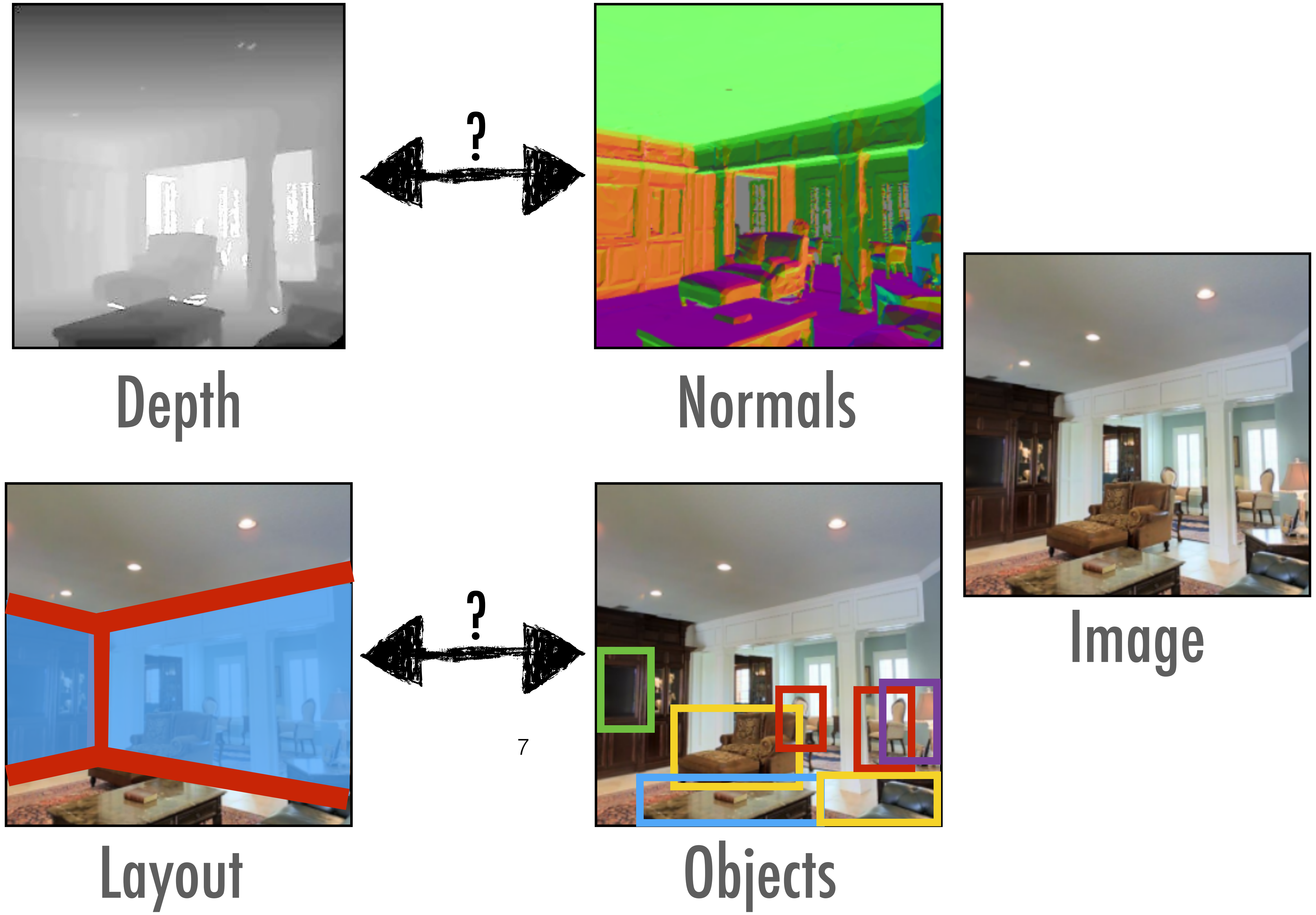
What is Taskonomy?

- From literal meaning: **Taskonomy** = **Task** + **Taxonomy**
- From Definition: Taskonomy is a research that **quantifies** the **relationship between different vision problems** and uses these relationship to optimize the training policy

So what is **Taskonomy**?

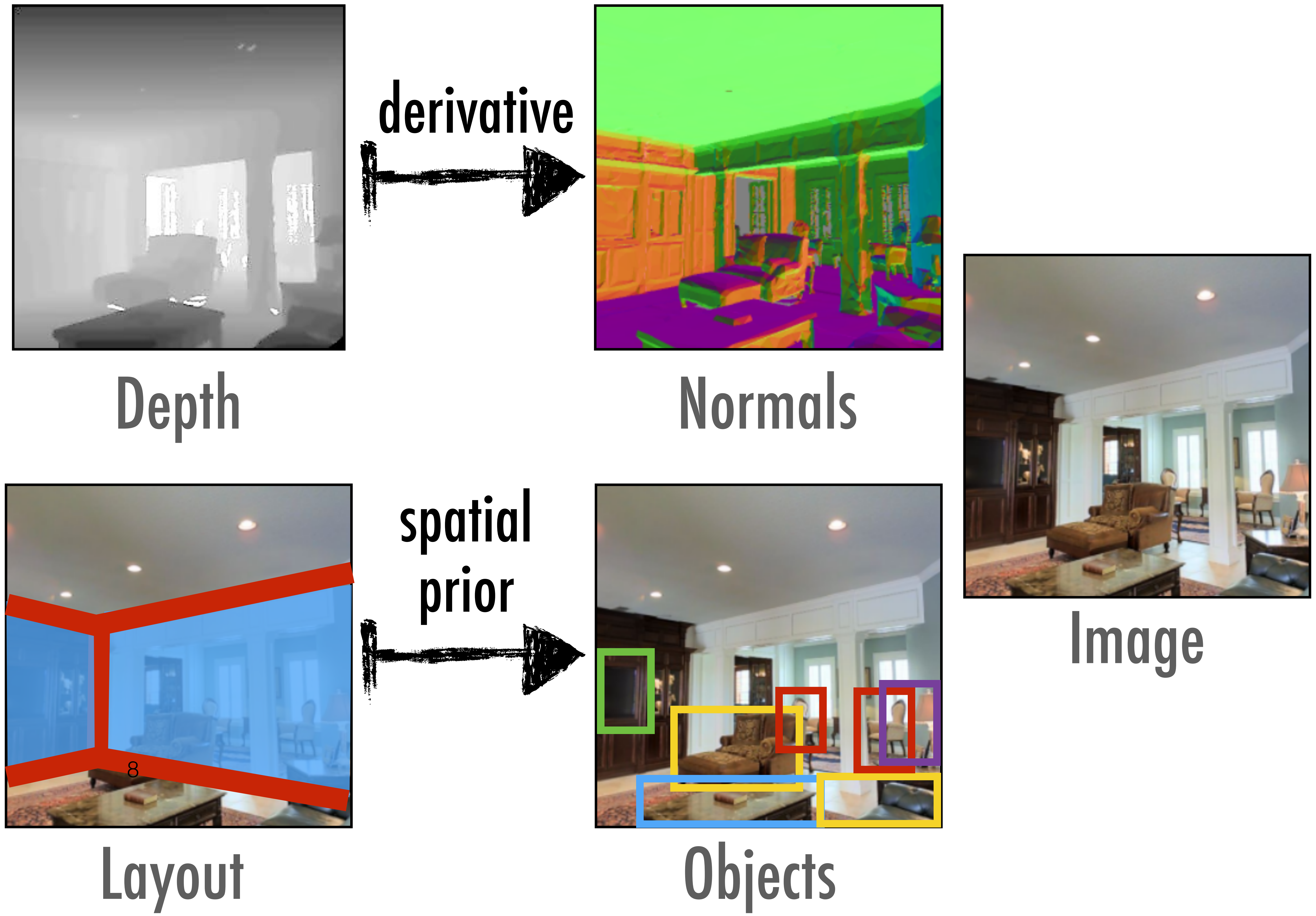
First, let us talk about the **relationship between Vision problems**

Question: Vision problems - related or independent?



Question: Vision problems - related or independent?

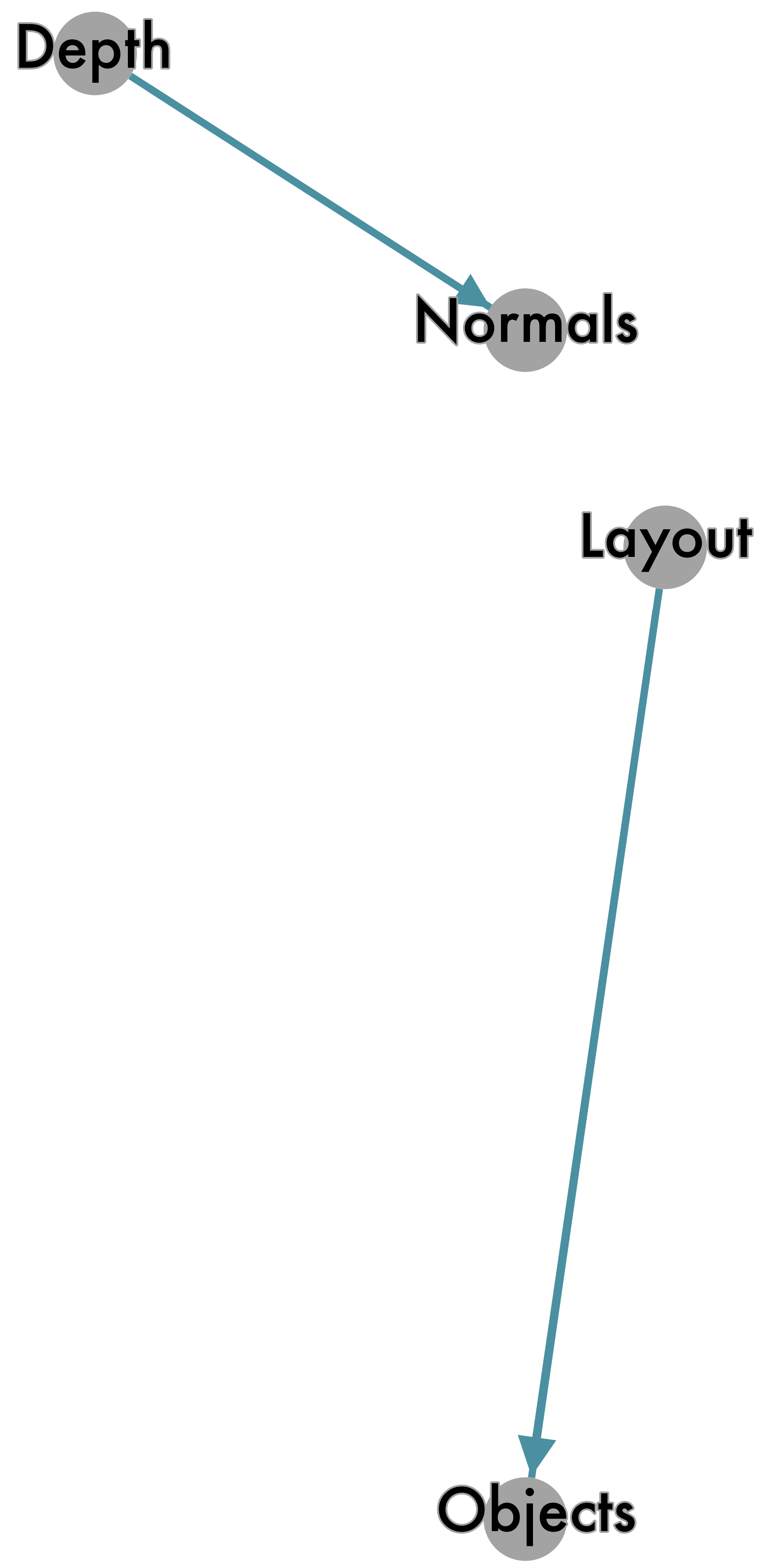
- Task relationships exist
- Can be computationally measured
- Tasks belong to a structured space
- Unified model for transfer learning



So how can we describe the relationship?

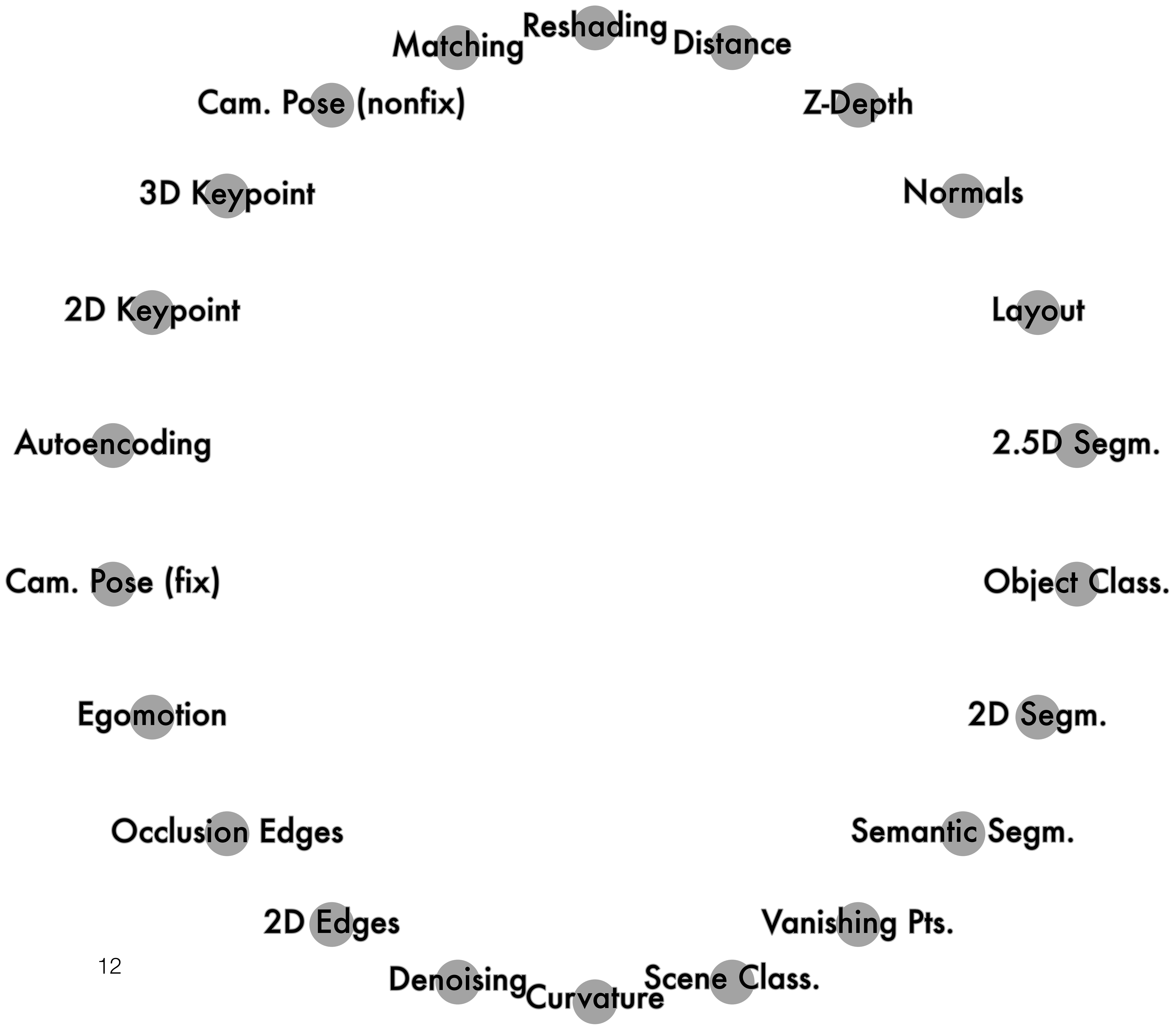
May be we can use a graph

Task Relationships



Task Relationships

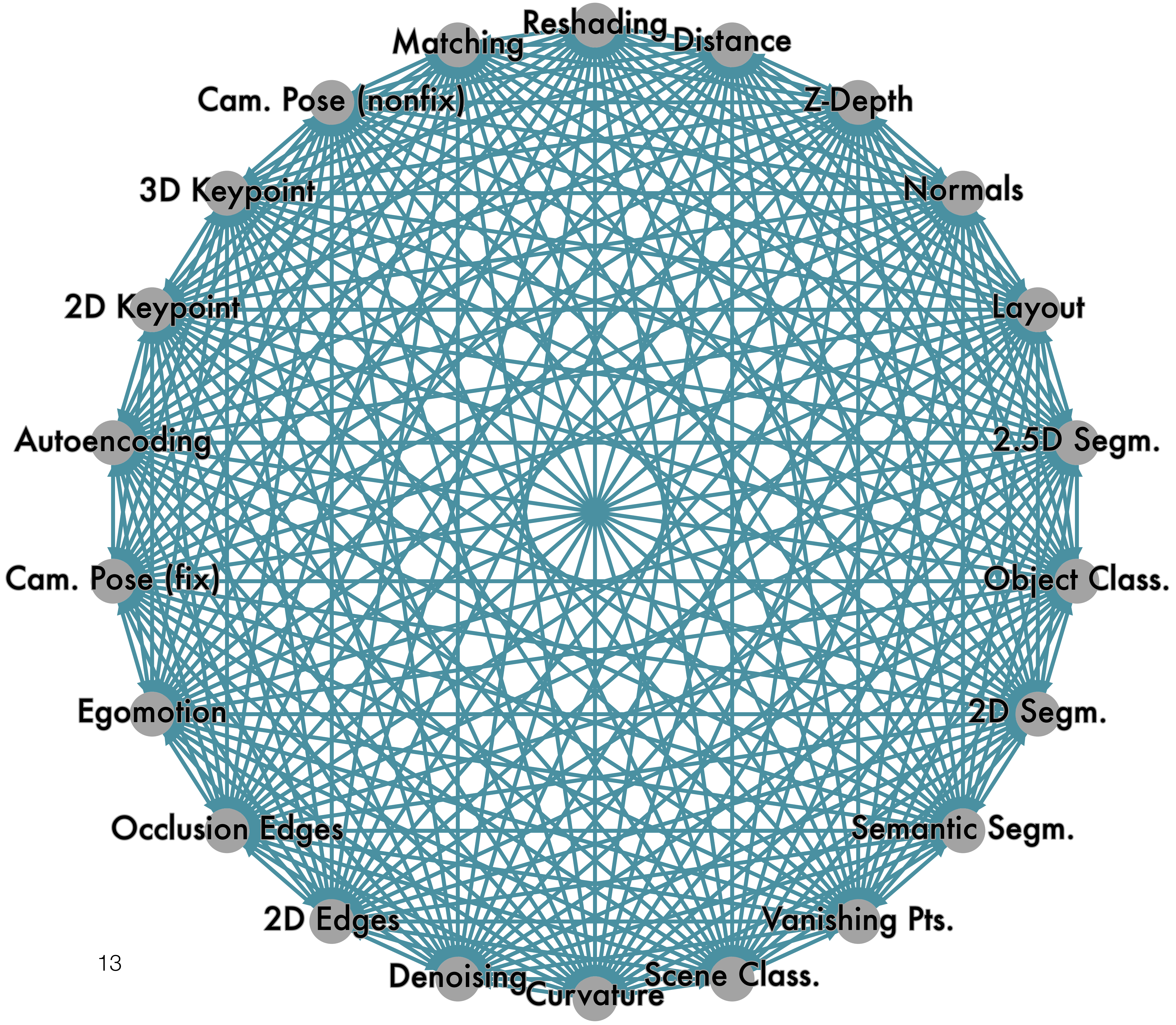
Many Tasks



Task Relationships

Many Tasks

Dense Relationships

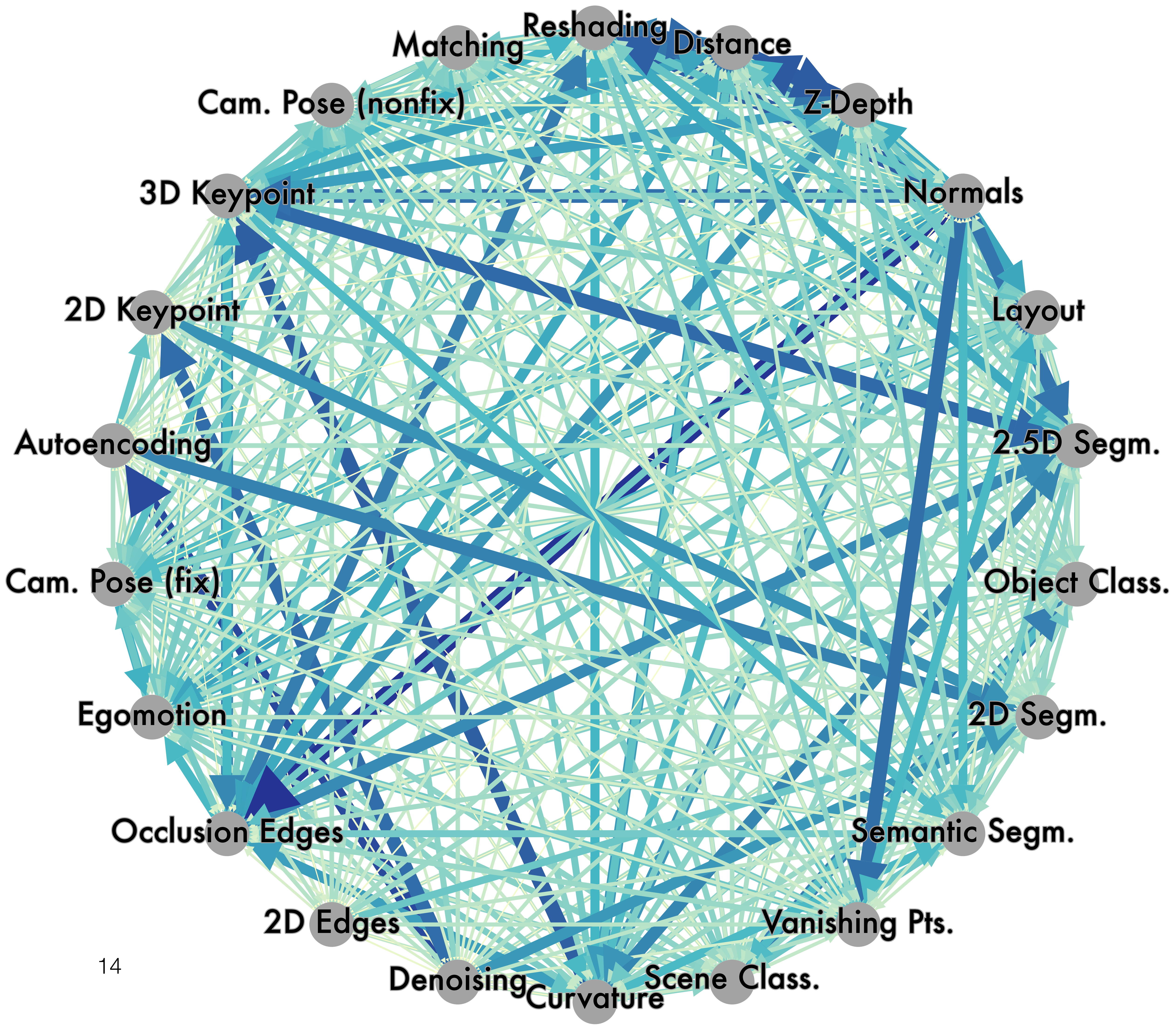


Task Relationships

Many Tasks

Dense Relationships

Redundancy



Task Relationships

Many Tasks

↓

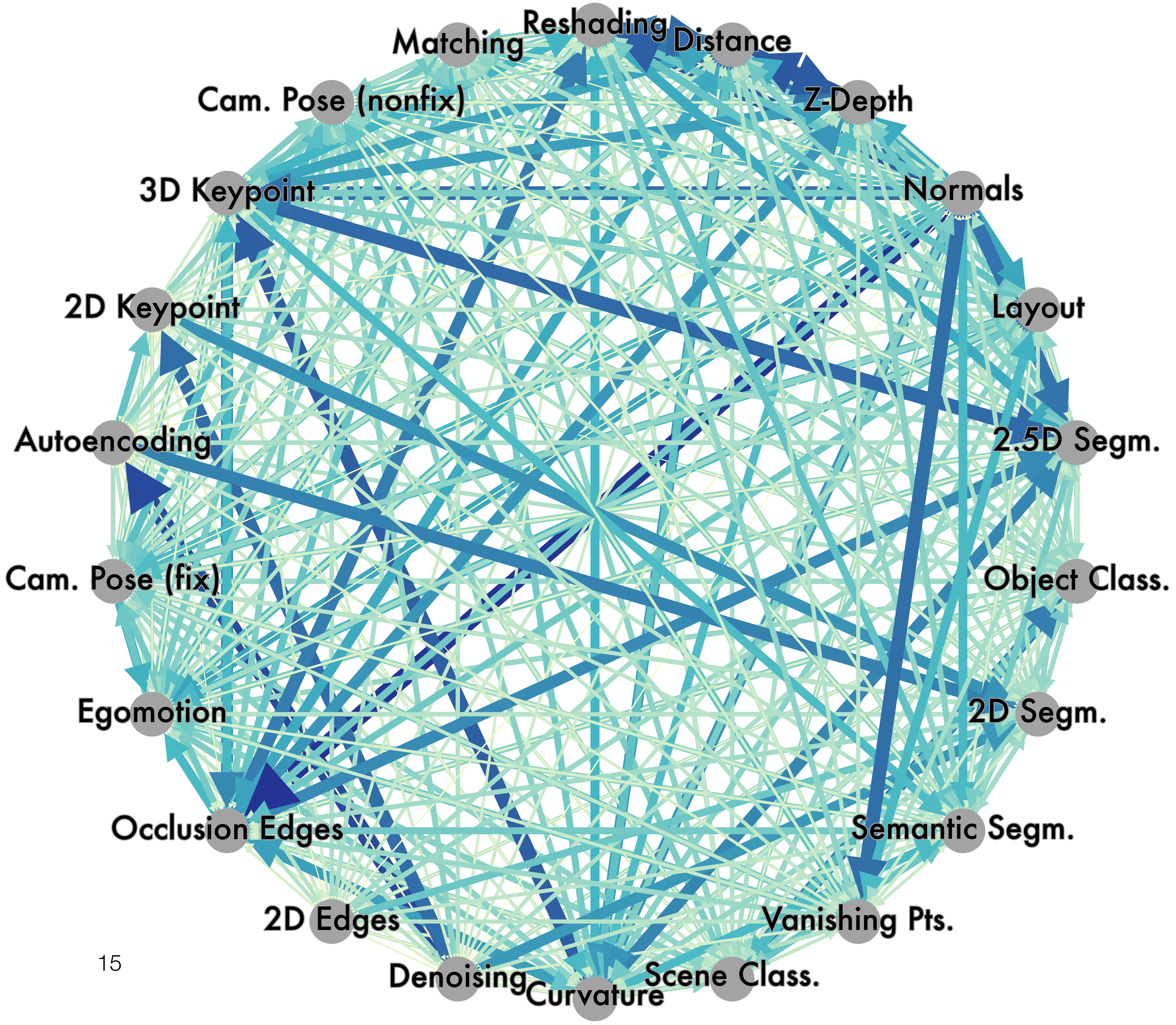
Dense Relationships

↓

Redundancy

↓

Efficiency



Task Relationships

Many Tasks

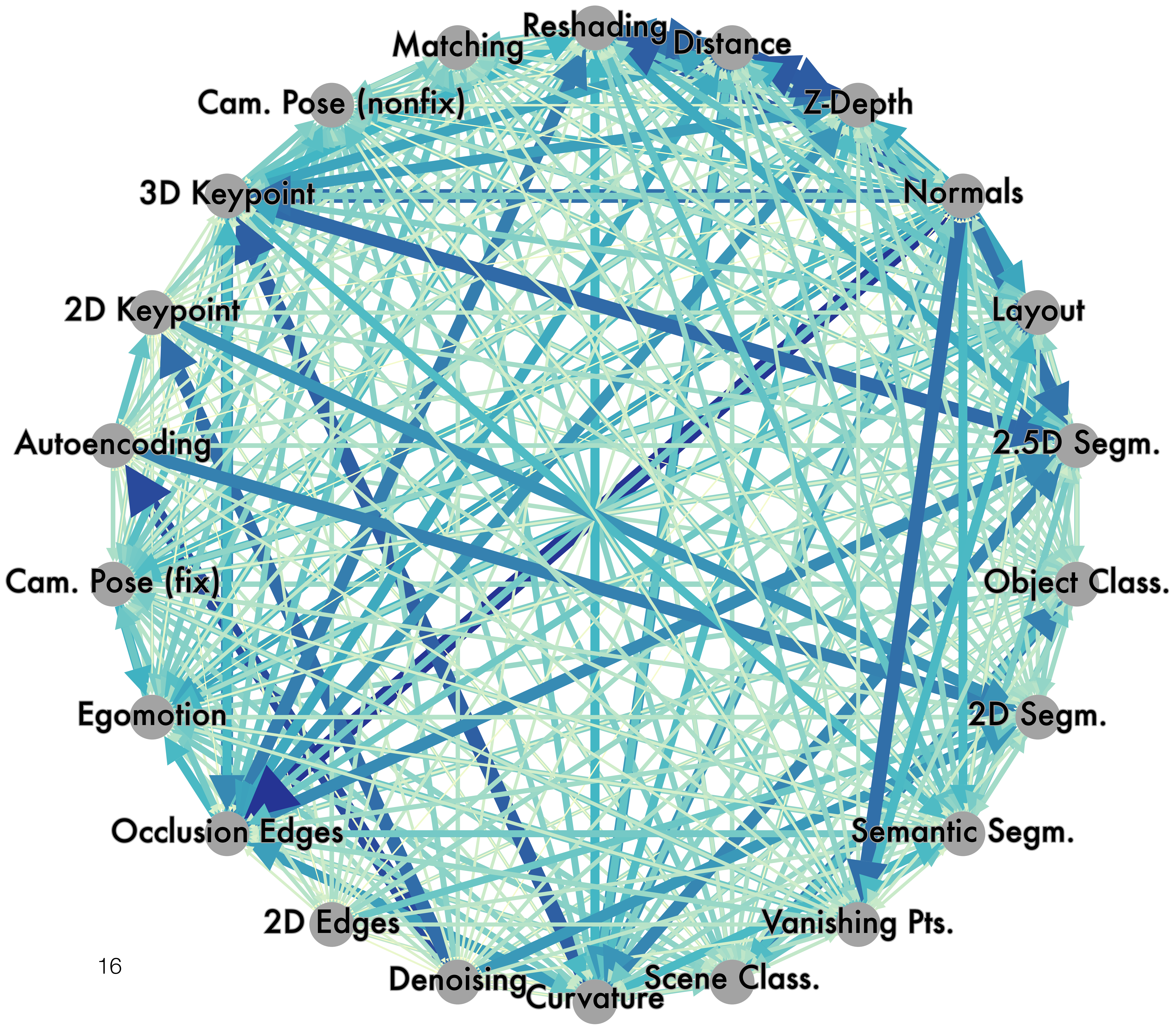
Dense Relationships

Redundancy

Efficiency

Supervision Efficiency

- Self/Un Supervised Learning [1,2,3]
- ImageNet features. “Fine-Tuning” [4,5,6]
- Meta Learning [7,8,9]
- Domain Adaptation [10,11,12]



[1] Doersch’15. [2] Noroozi’16. [3] Bengio’13. [4] Sharif’14. [5] Donahue’14. [6] Girshick’14. [7] Marcin’14. [8] Finn’16. [9] Lake’16. [10] Yang’07. [11] Saenko’10. [12] Hoffman’14.

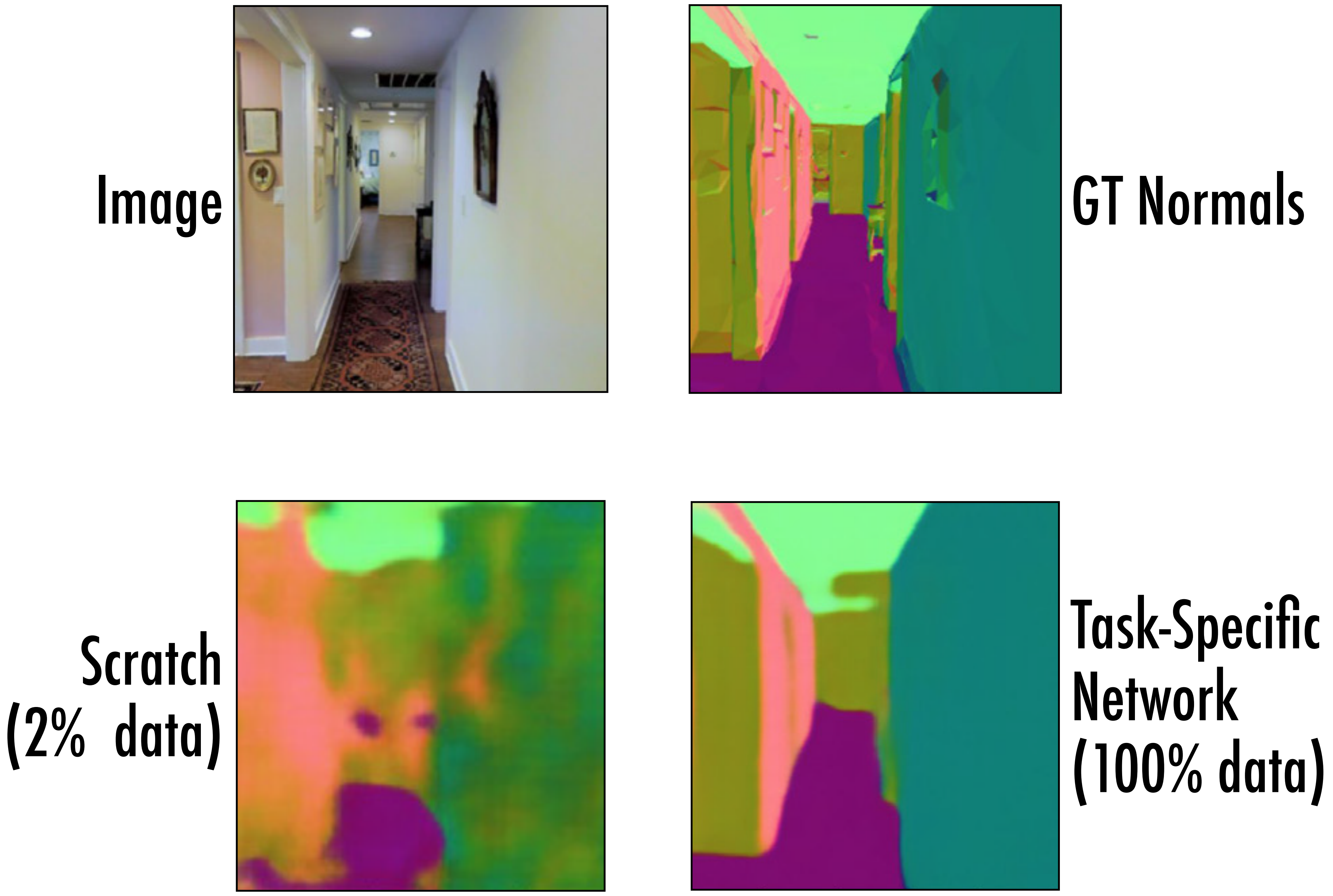
So how can we use the relationship?

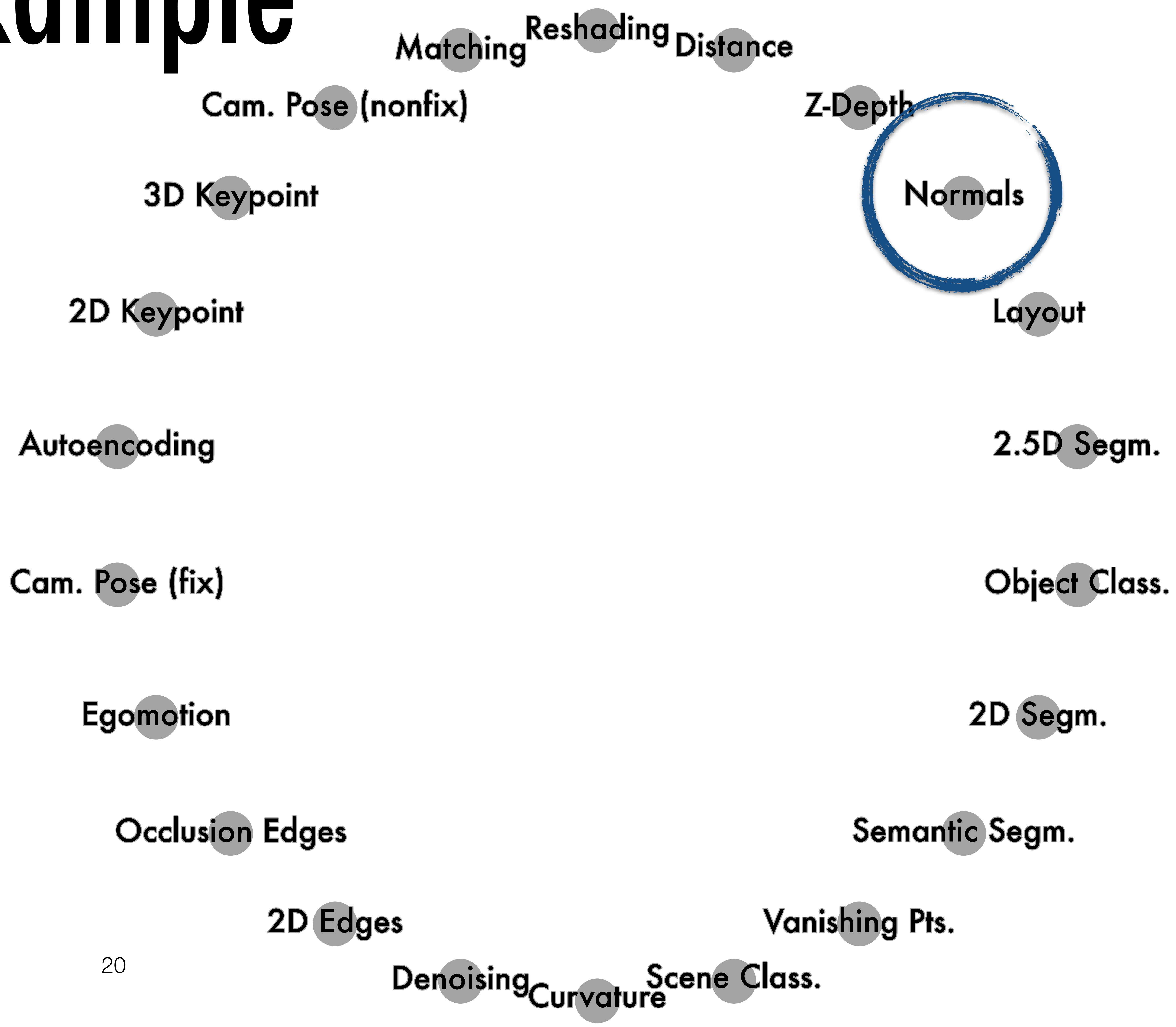
We can use it in **transfer learning**

What is transfer learning

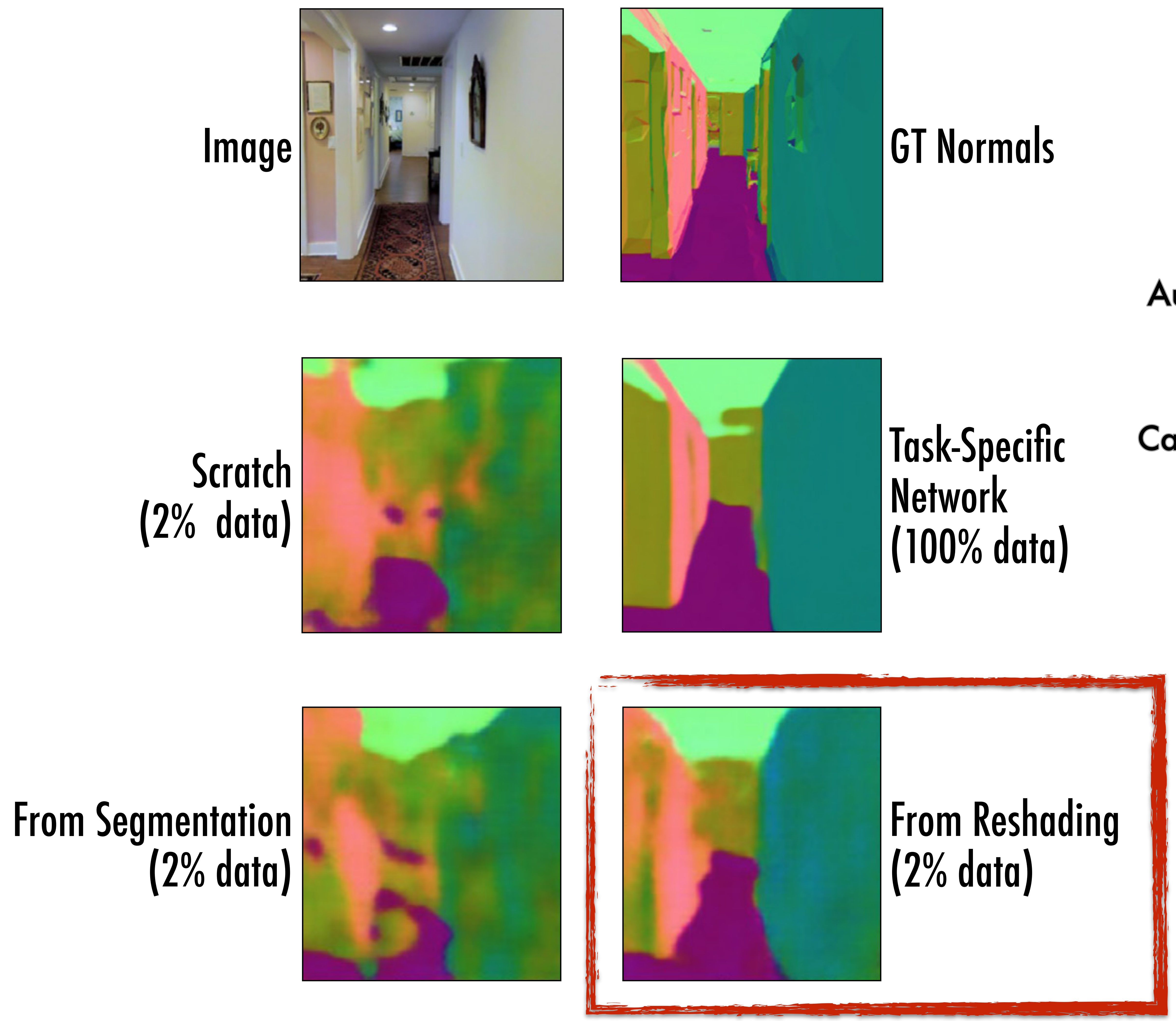
- Model developed for task X may be useful for solving task Y, if <u>X and Y related</u>.
- So transfer learning is used to solve task Y by using the arguments in the model which is used to solve task X
- For example, if we use a neural network A to solve task X, we can use some of the arguments in it to solve task Y if they are related

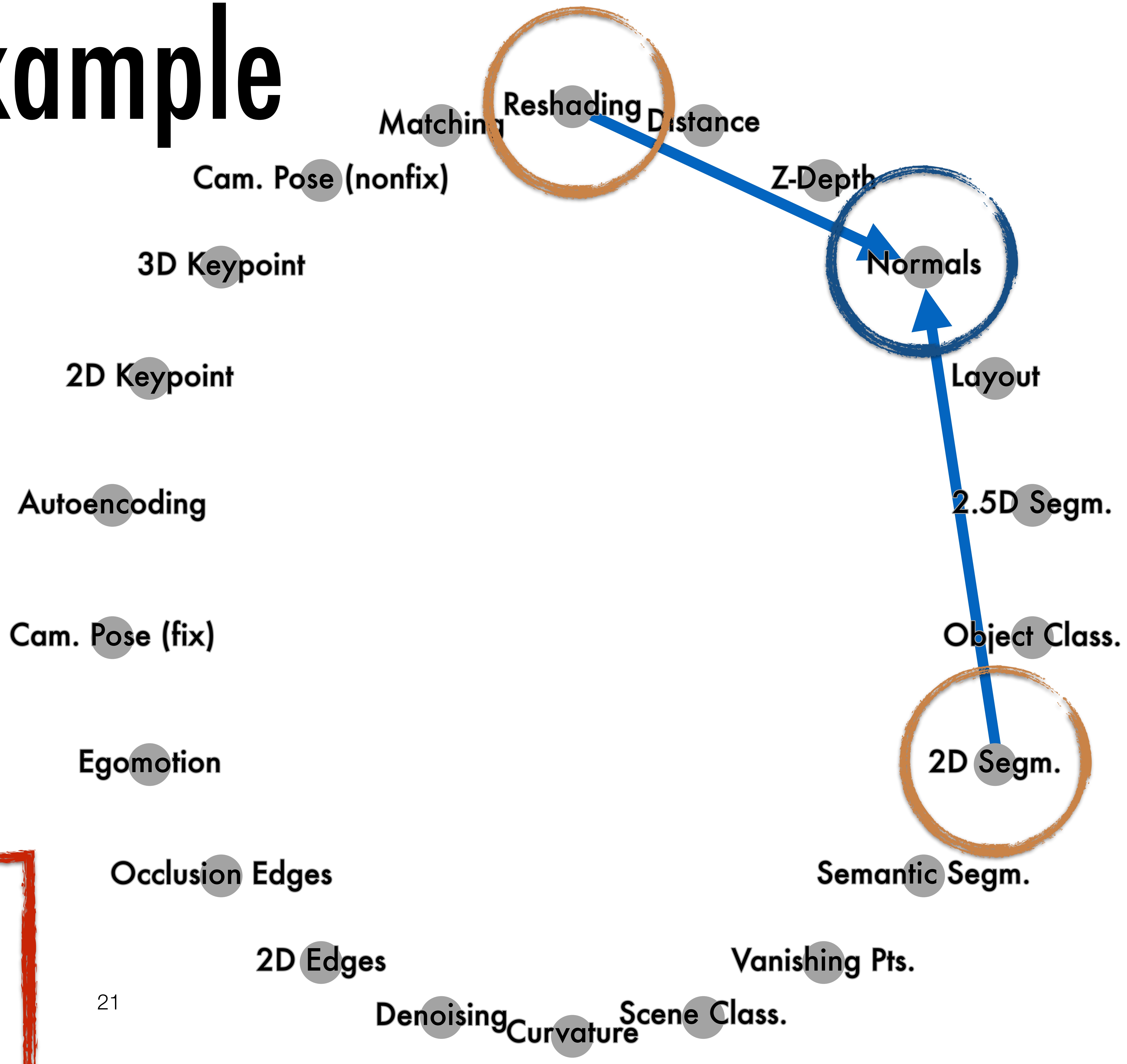
Transfer Learning Example



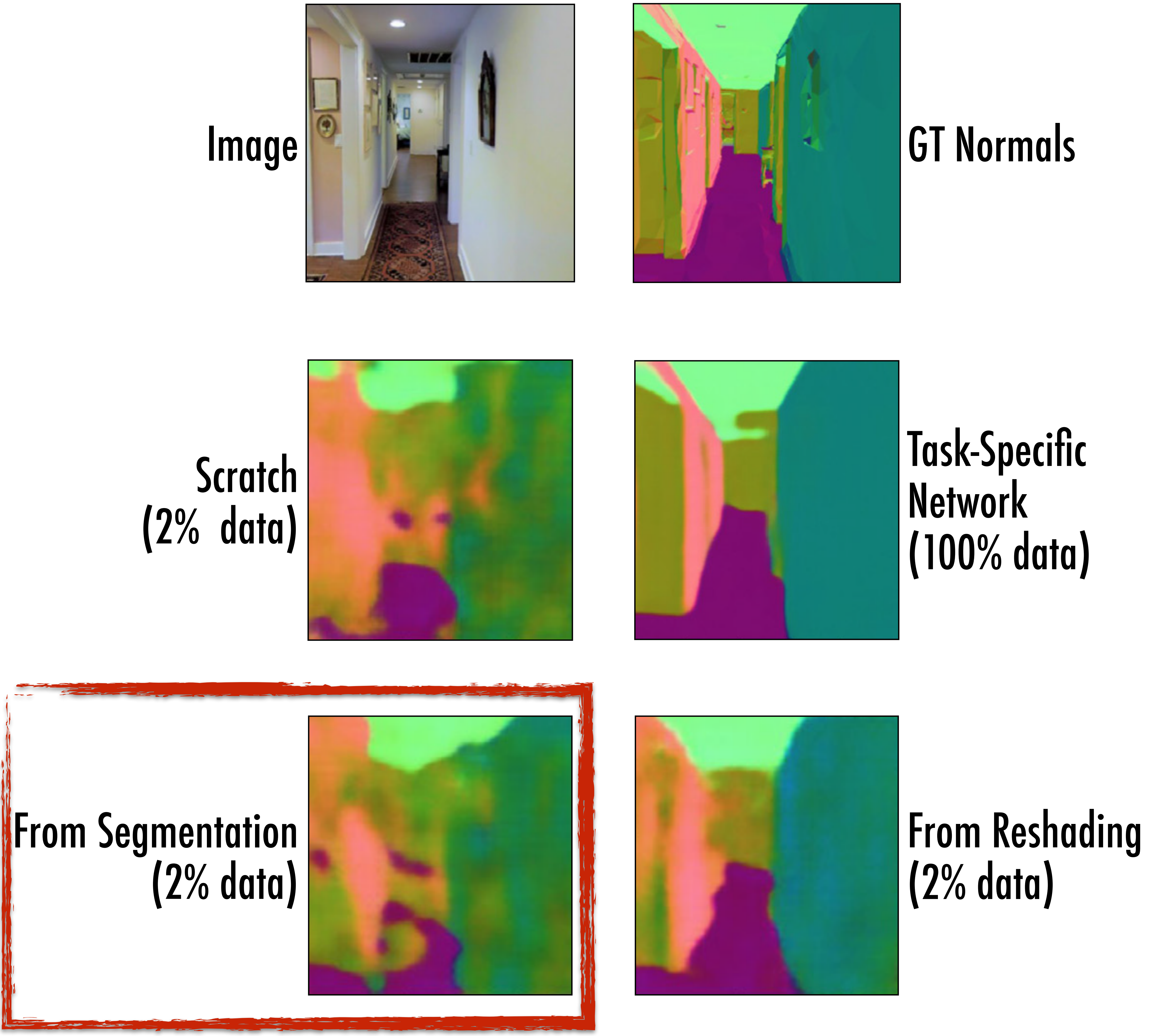


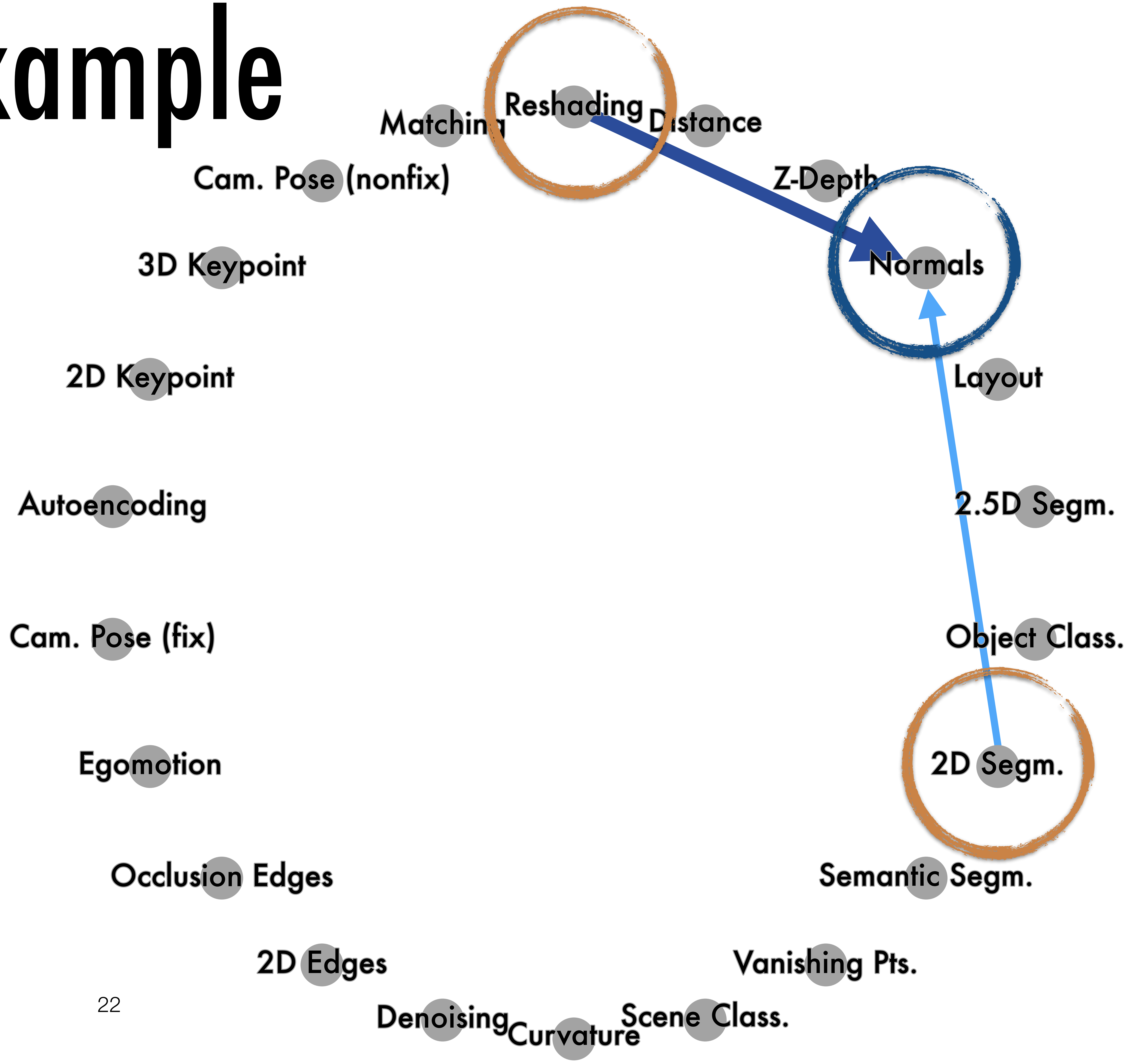
Transfer Learning Example





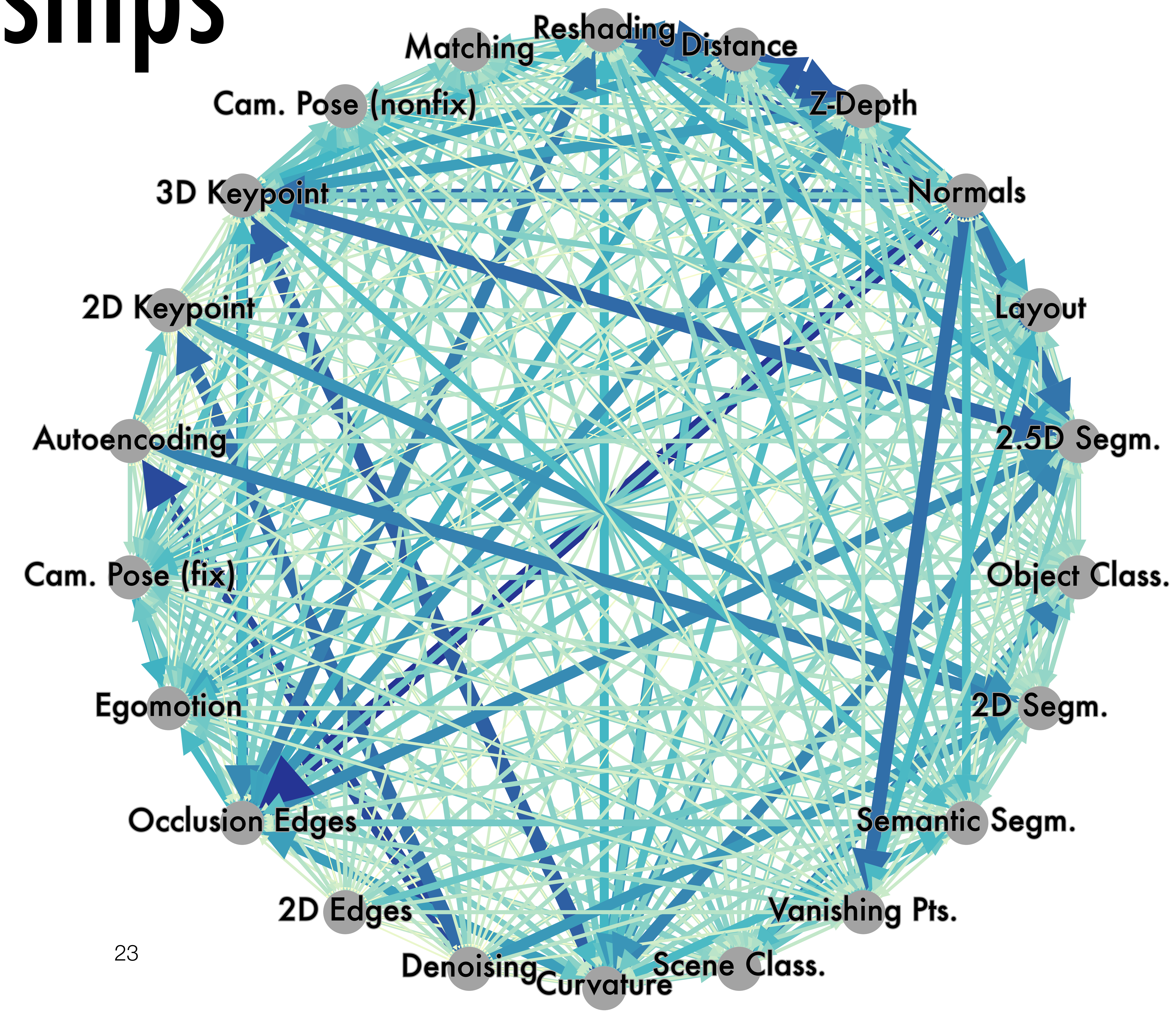
Transfer Learning Example





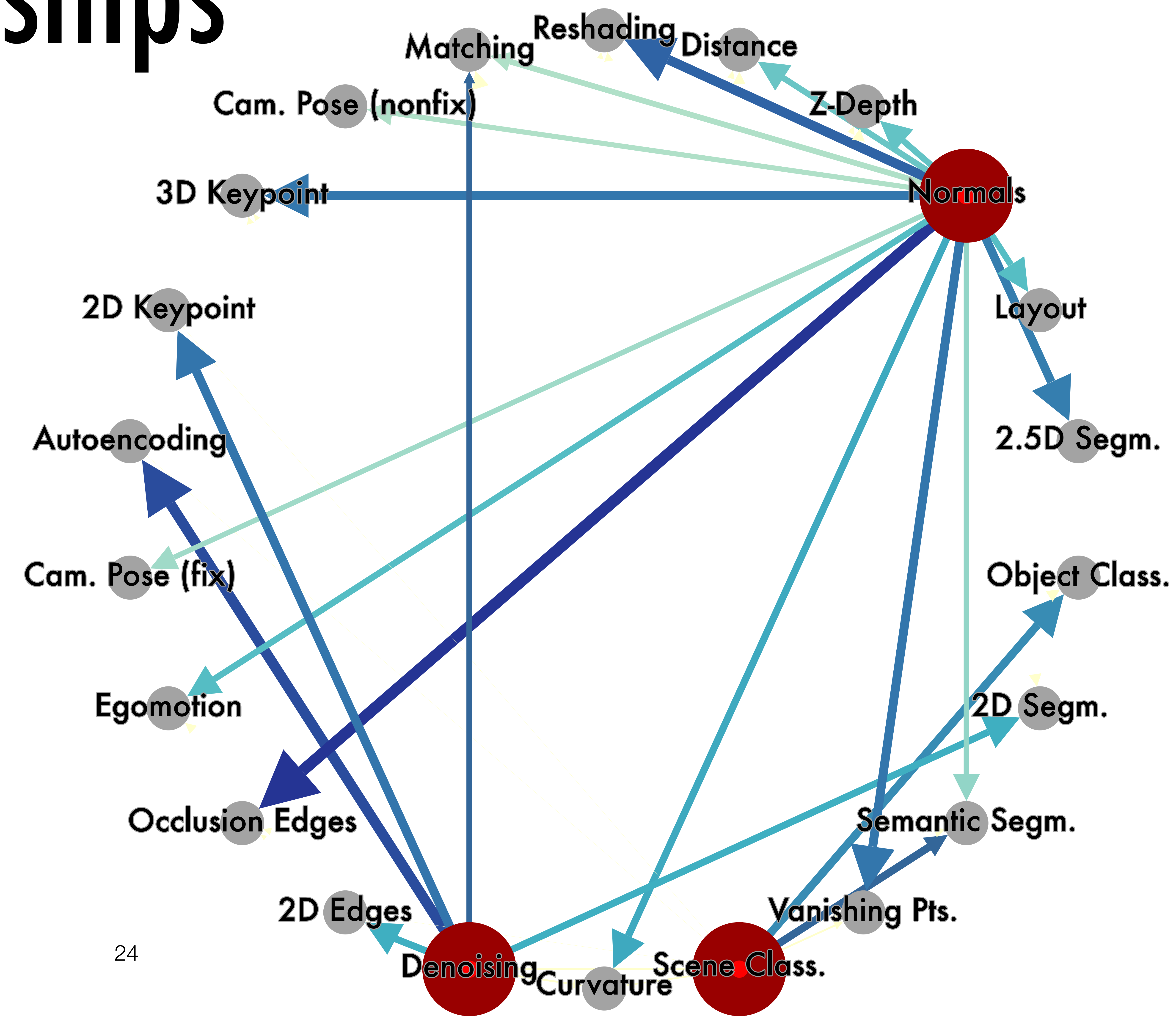
Dense Task Relationships

- Global window to redundancies



Dense Task Relationships

- Global window to redundancies
- Devise a policy to solve a set of tasks
  - in concert
  - recycle supervision

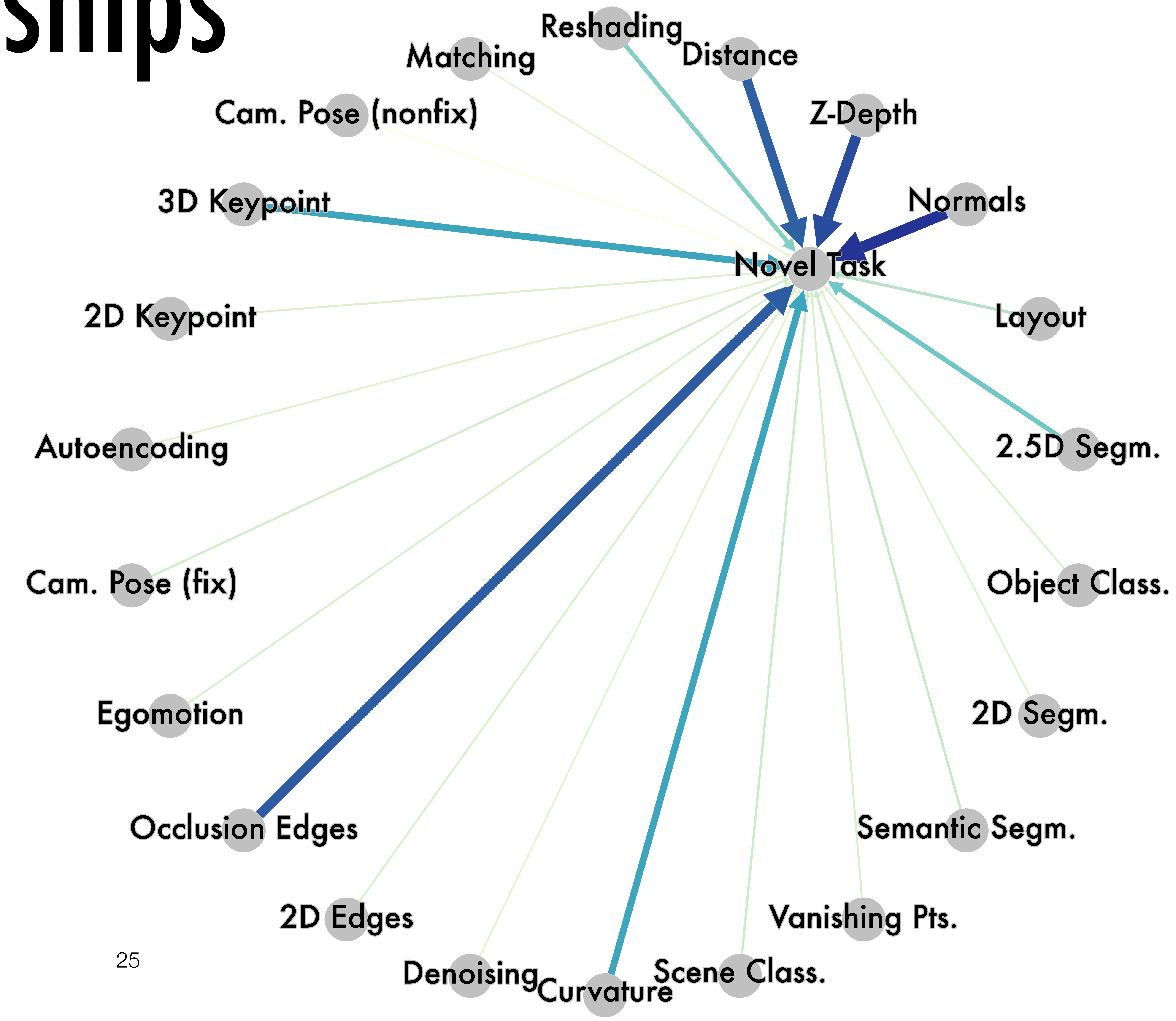


Dense Task Relationships

- Global window to redundancies

- Devise a policy to solve a set of tasks
  - in concert
  - recycle supervision

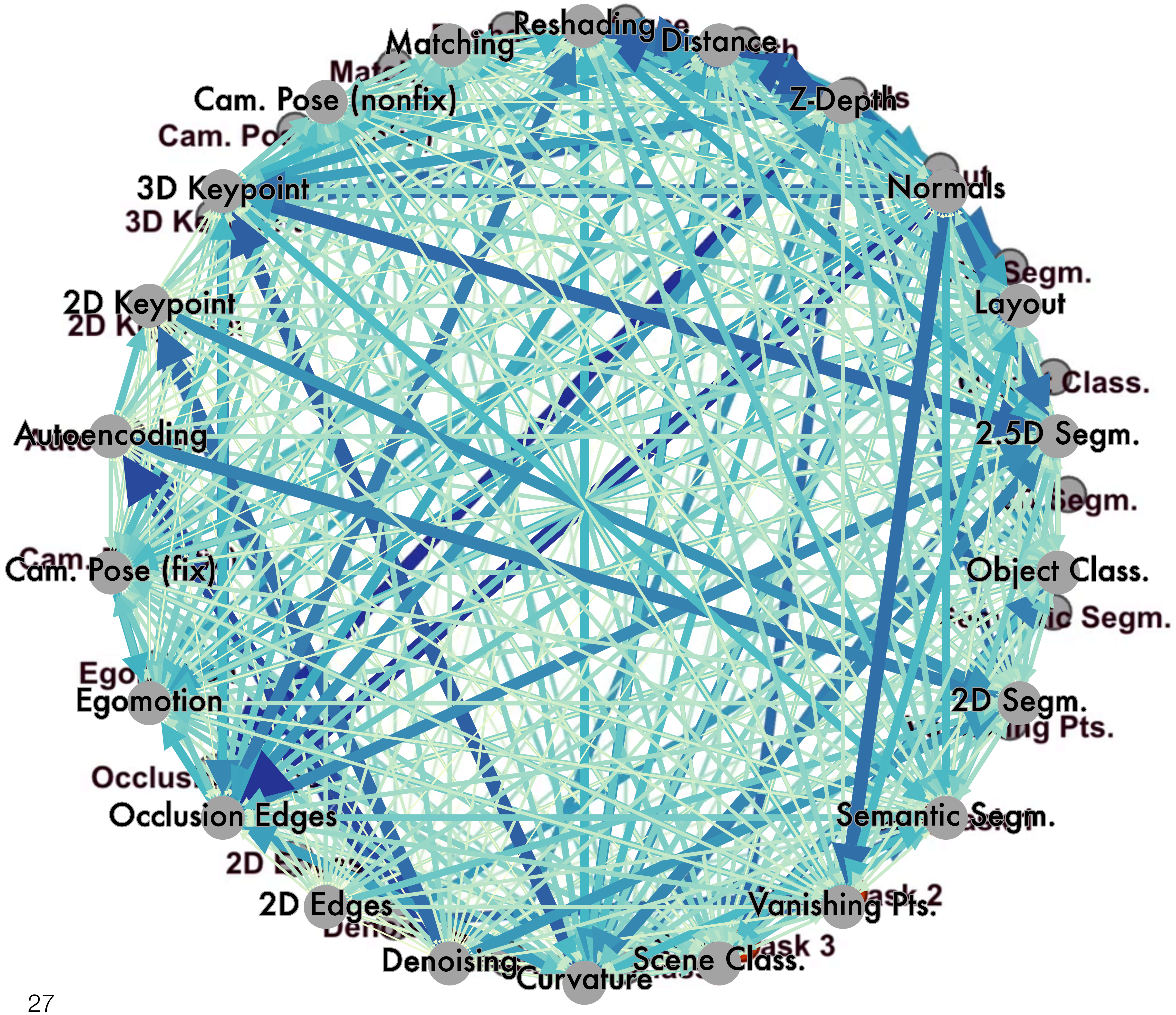
- Solve a novel task with little data
  - by “insertion” into this structure



So what is **Taskonomy**?

Taskonomy

- A fully computational method for quantifying task relationships
- Extracting a structure out of them
- Unified model for transfer learning
- **task** tax**onomoy** ≈ taskonomy



Why Taskonomy is needed?

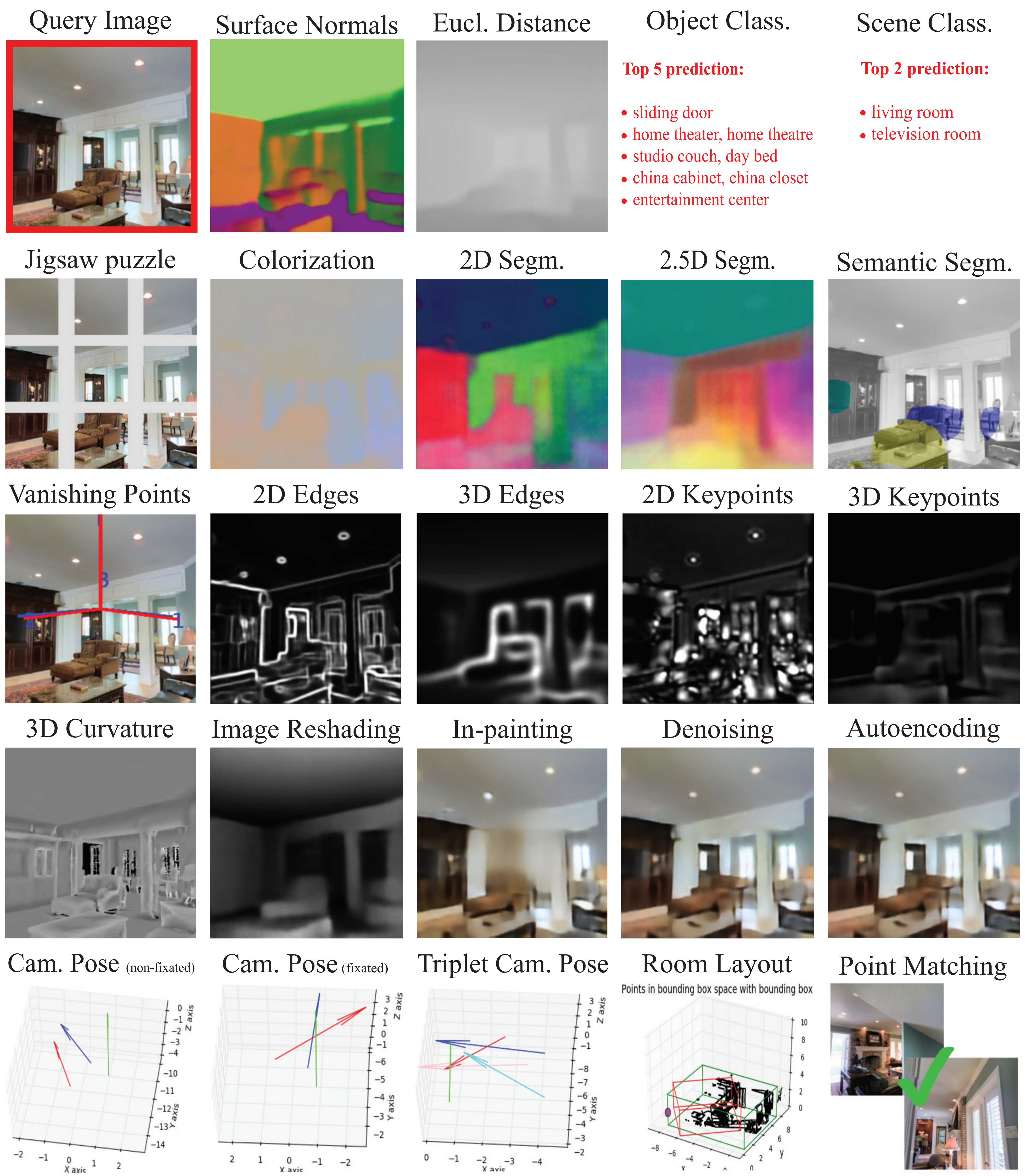
- The current methods in computer vision has gone far without explicitly using the relationships. Most of them solve the problems one by one. So they make two problems
  - 1. Solving the vision problem one by one should collect a lot of data for each task, with the growing of the number of task, it will be infeasible
  - 2. Solving different vision problems one by one will bring redundant computation and repeated learning between different tasks that will cost too much computing resource
- Taskonomy quantifies the relationship between related tasks so we can use transfer learning to learn different task using less data and saves more computing resource and from the paper, the main works which Taskonomy is used do is:
  - 1.Using it to solve a group tasks
  - 2.Using its task dictionary to solve some problems with little data

Contents

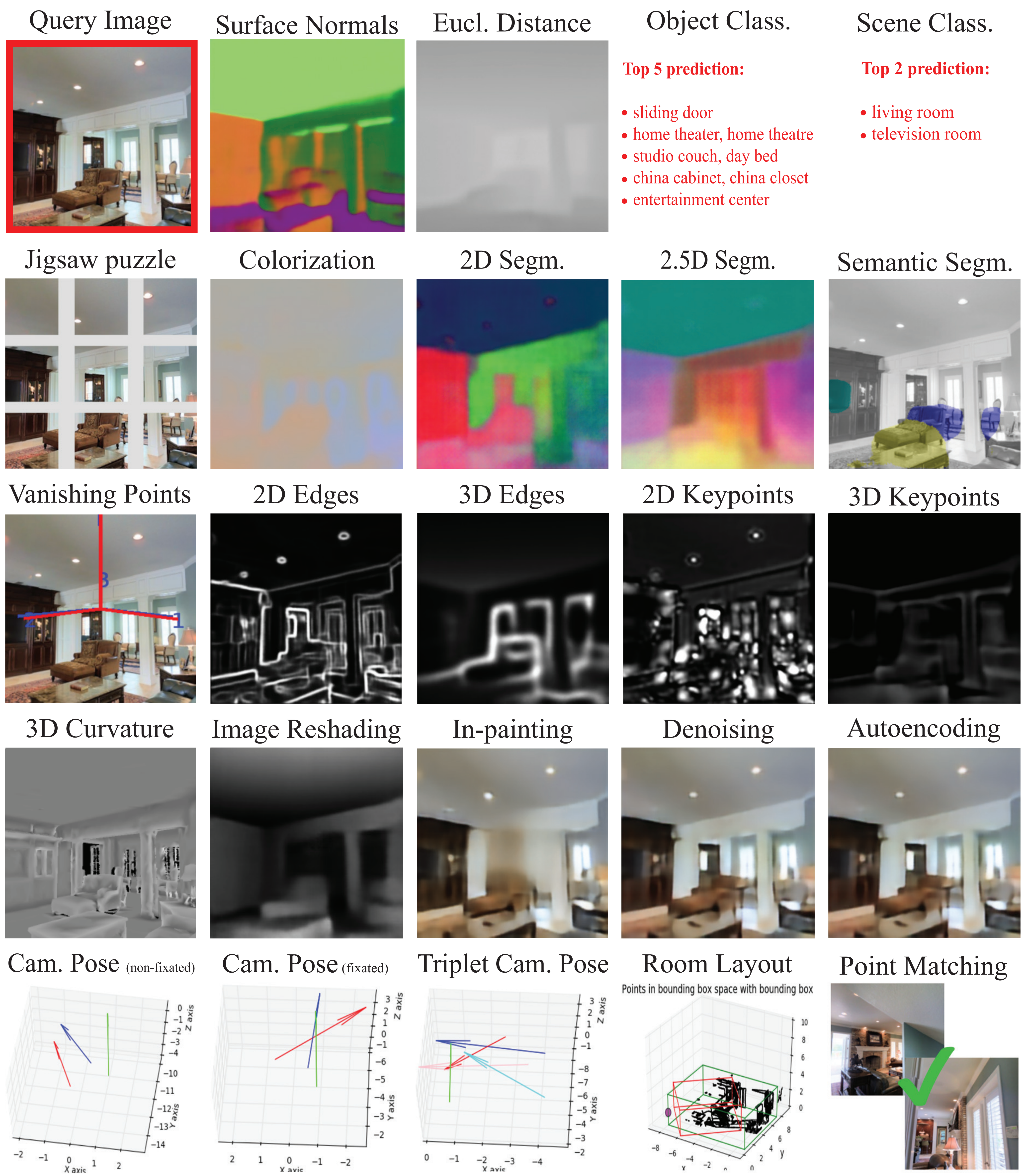
- Introduction
- Method
- Result and Summary

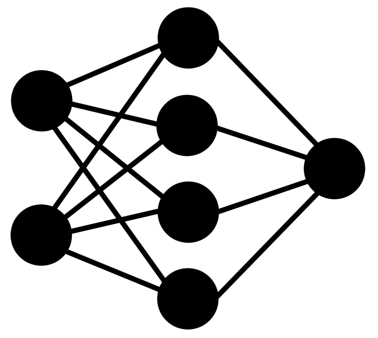
Pre works of the training – task bank, data set and Task-Specific Networks

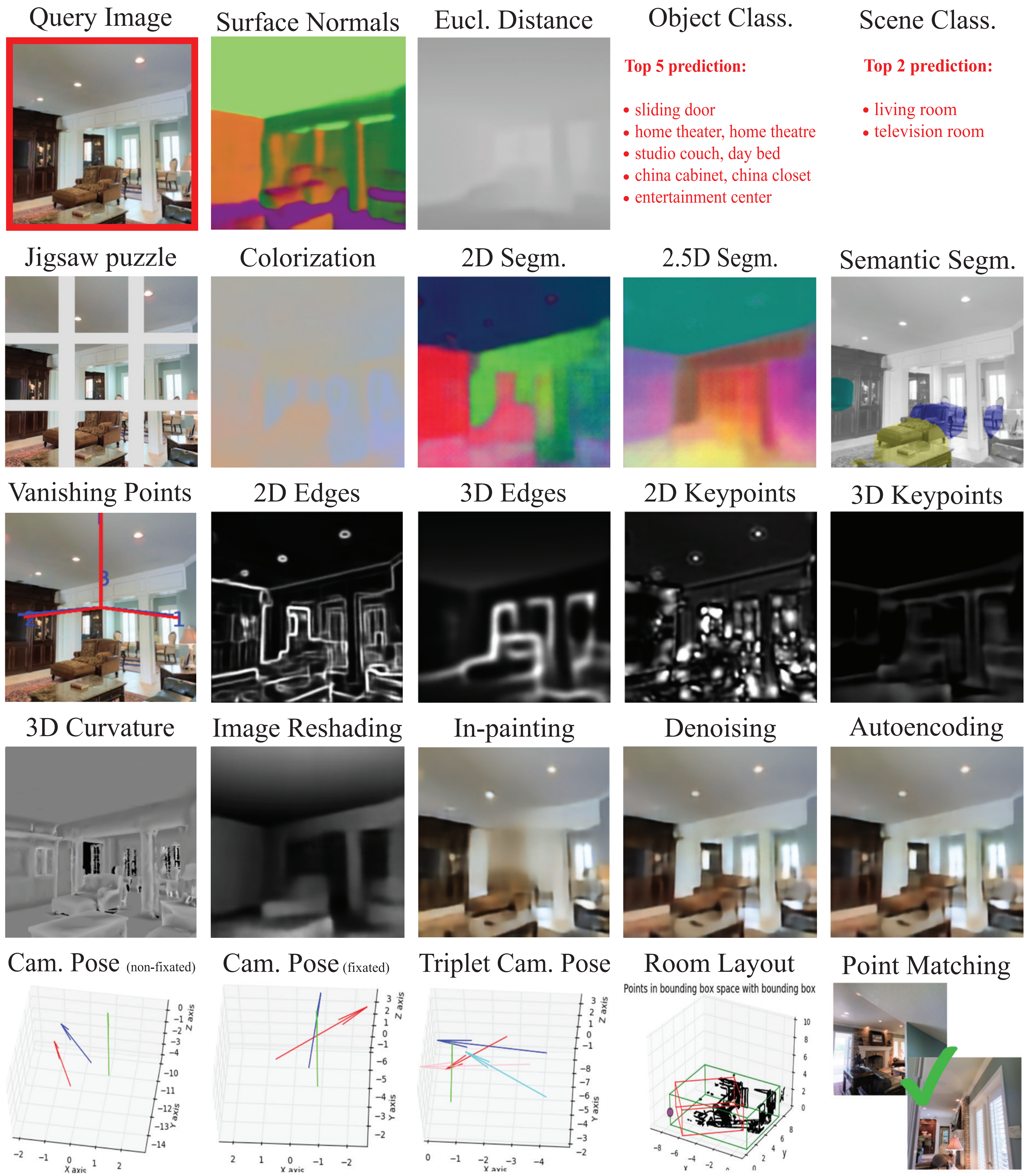
- **Task Bank**
    - 26 Semantic, 2D and 3D tasks



- **Task Bank**
  - 26 Semantic, 2D and 3D tasks
- **Dataset**
  - 4 million real images
  - Each image has the GT label for all tasks



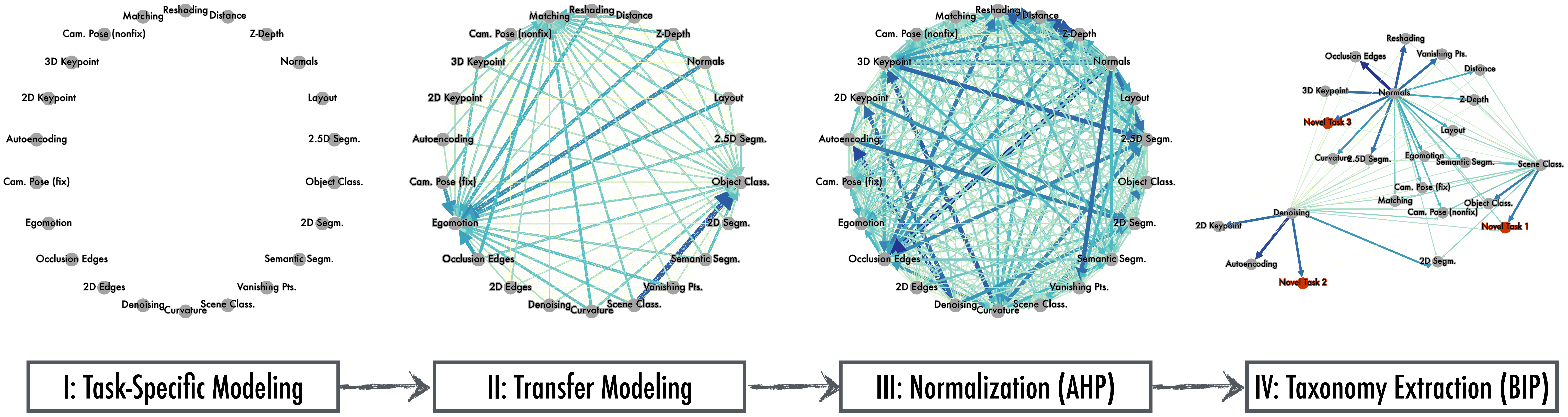
- **Task Bank**
  - 26 Semantic, 2D and 3D tasks
- **Dataset**
  - 4 million real images
  - Each image has the GT label for all tasks
- **Task-Specific Networks**
  - 26 x 



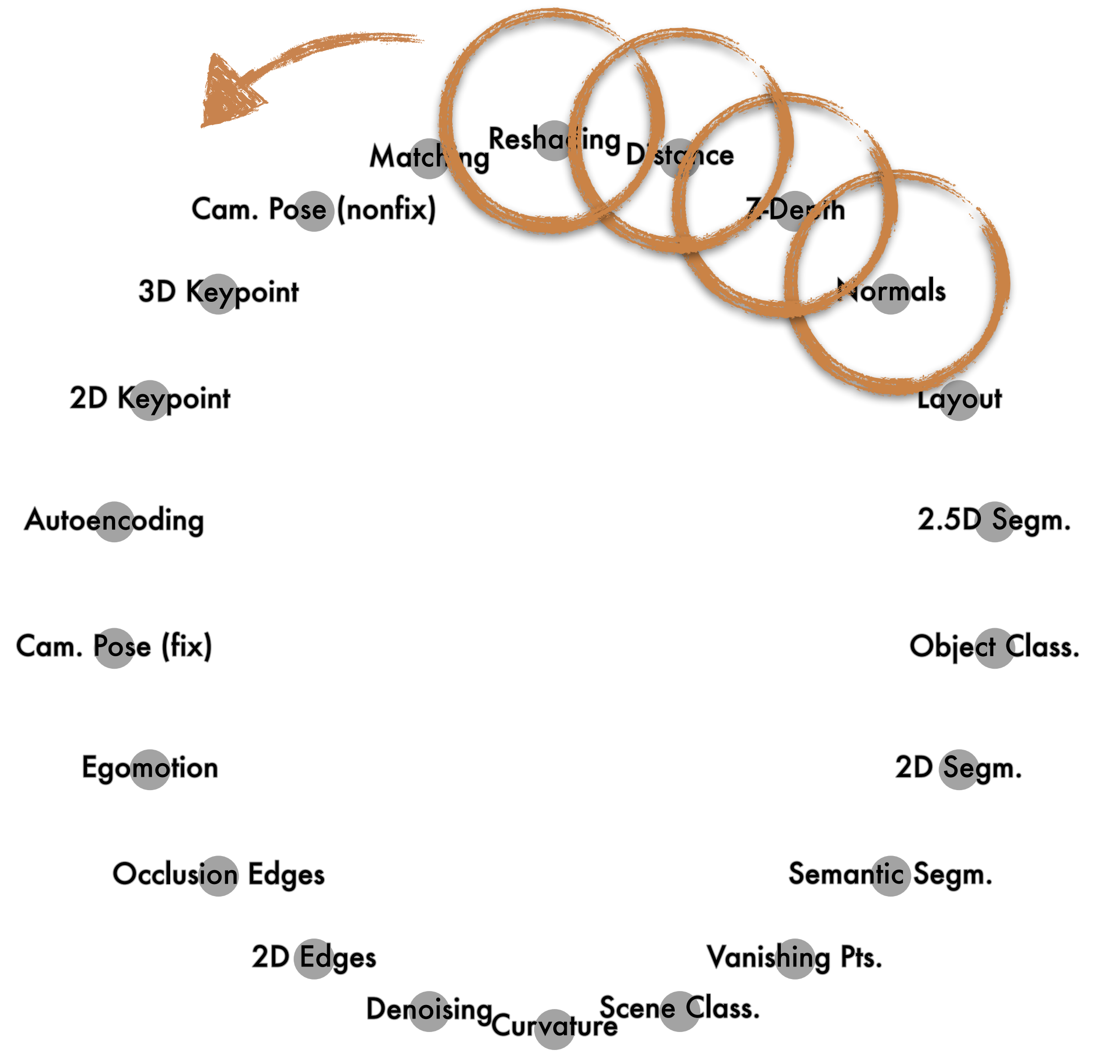
Four steps of the modeling process

- **1.** Task-Specific Modeling
- **2.** Transfer Modeling
- **3.** Normalization (AHP)
- **4.** Taxonomy Extraction (BIP)

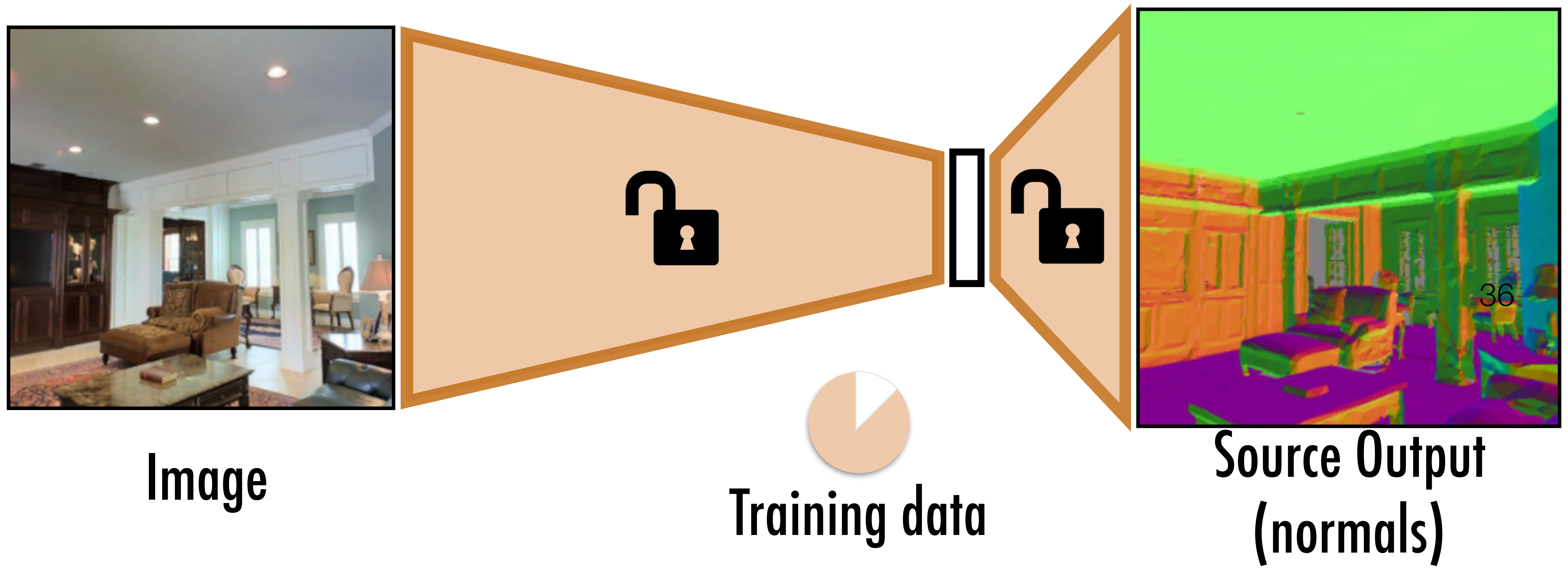
Modeling


Matching
Reshading
Distance
Cam. Pose (nonfix)
Z-Depth
3D Keypoint
Normals
2D Keypoint
Layout
Autoencoding
2.5D Segm.
Cam. Pose (fix)
Object Class.
Egomotion
2D Segm.
Occlusion Edges
Semantic Segm.
2D Edges
Vanishing Pts.
Denoising
Curvature
Scene Class.
Novel Task 1
Novel Task 2
Novel Task 3
I: Task-Specific Modeling
II: Transfer Modeling
III: Normalization (AHP)
IV: Taxonomy Extraction (BIP)

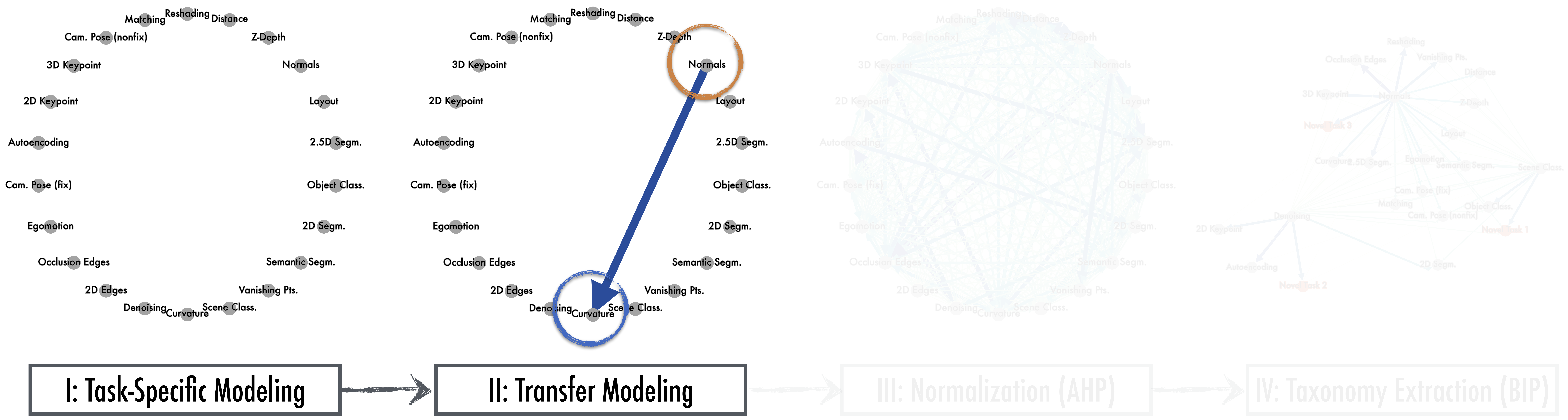
I: Task-Specific Modeling

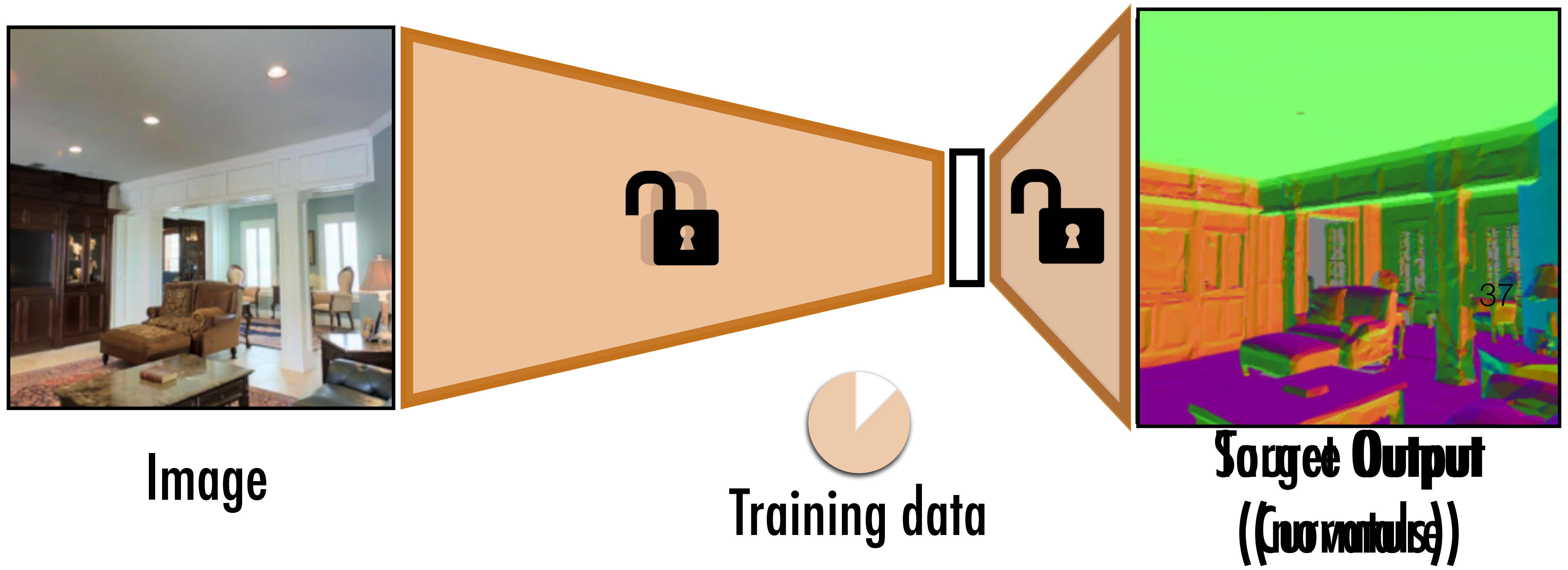


I: Task-Specific Modeling → II: Transfer Modeling → III: Normalization (AHP) → IV: Taxonomy Extraction (BIP)

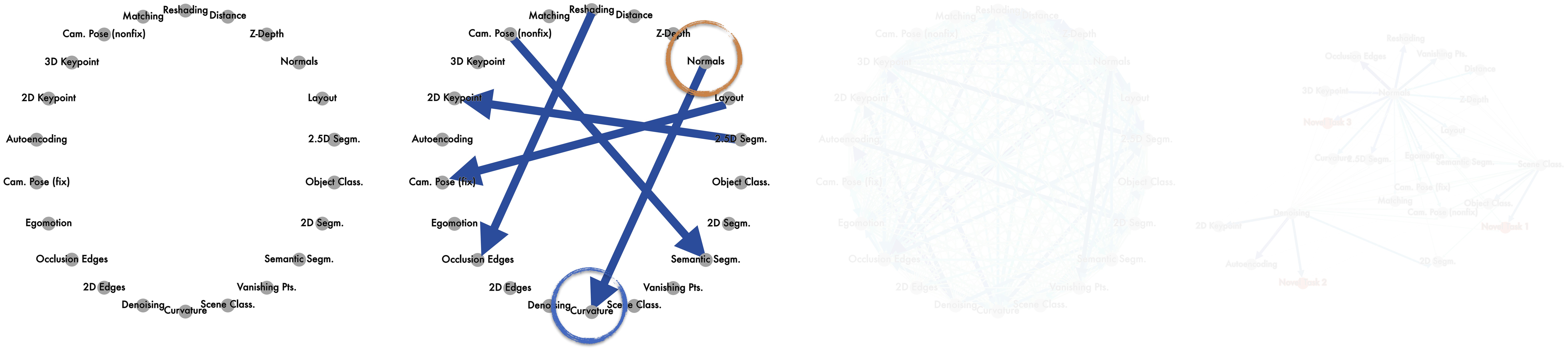


II: Transfer Modeling

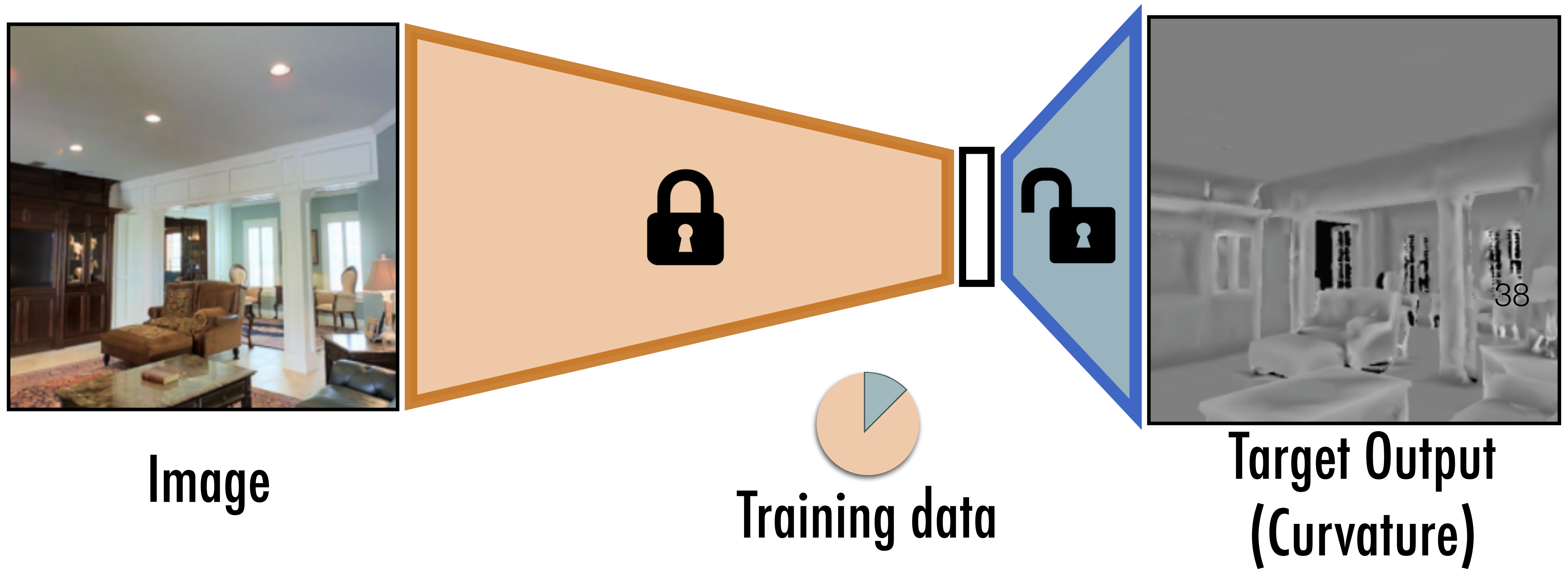




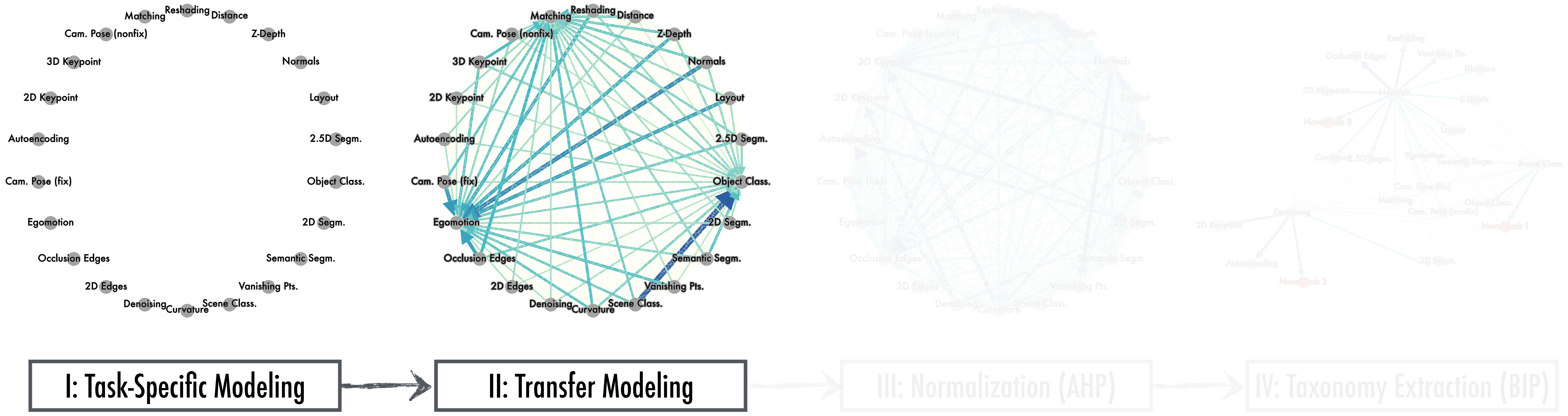
II: Transfer Modeling

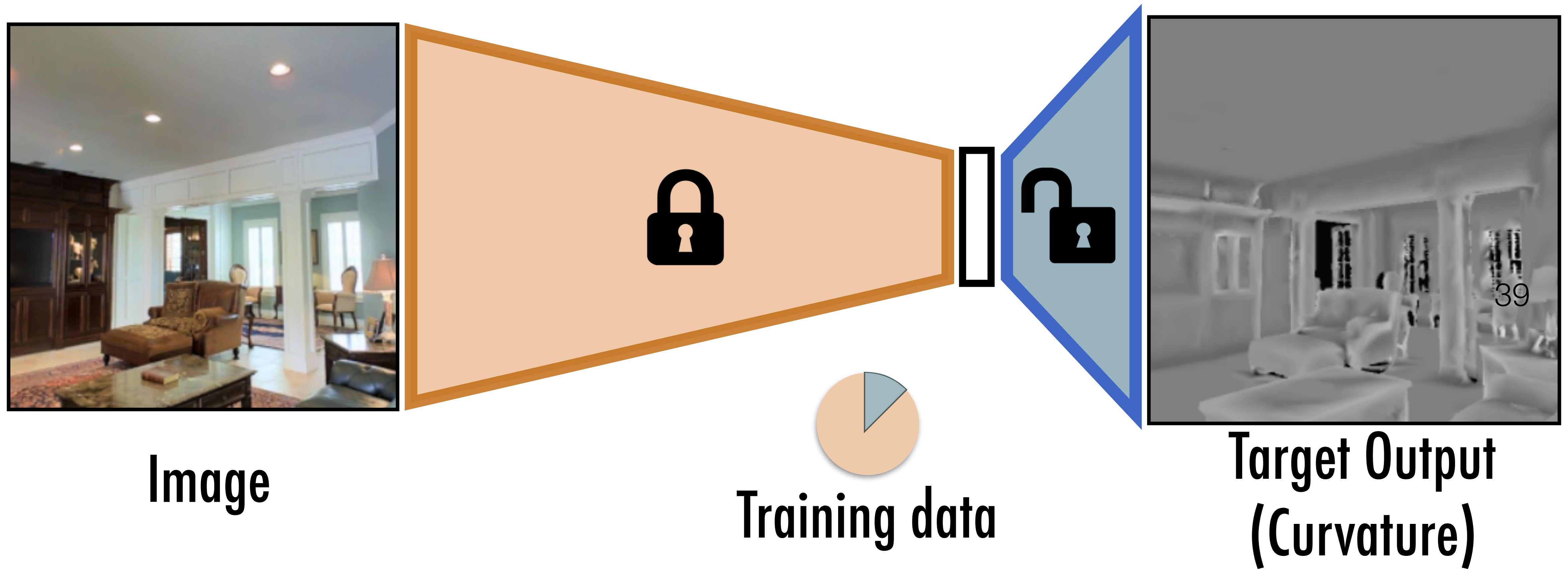


I: Task-Specific Modeling → II: Transfer Modeling → III: Normalization (AHP) → IV: Taxonomy Extraction (BIP)

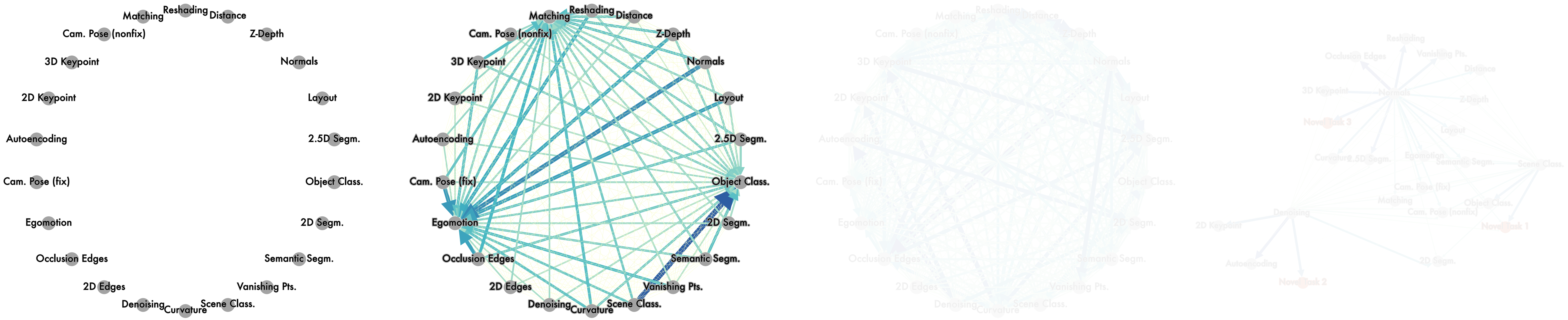


II: Transfer Modeling



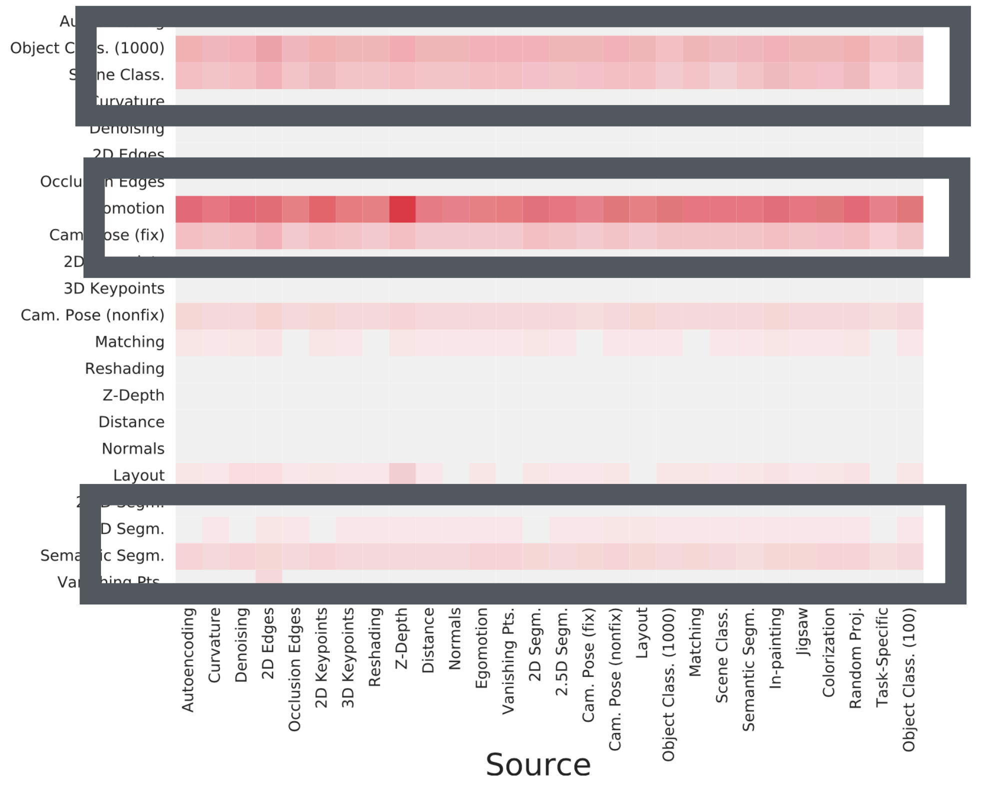


III: Normalization

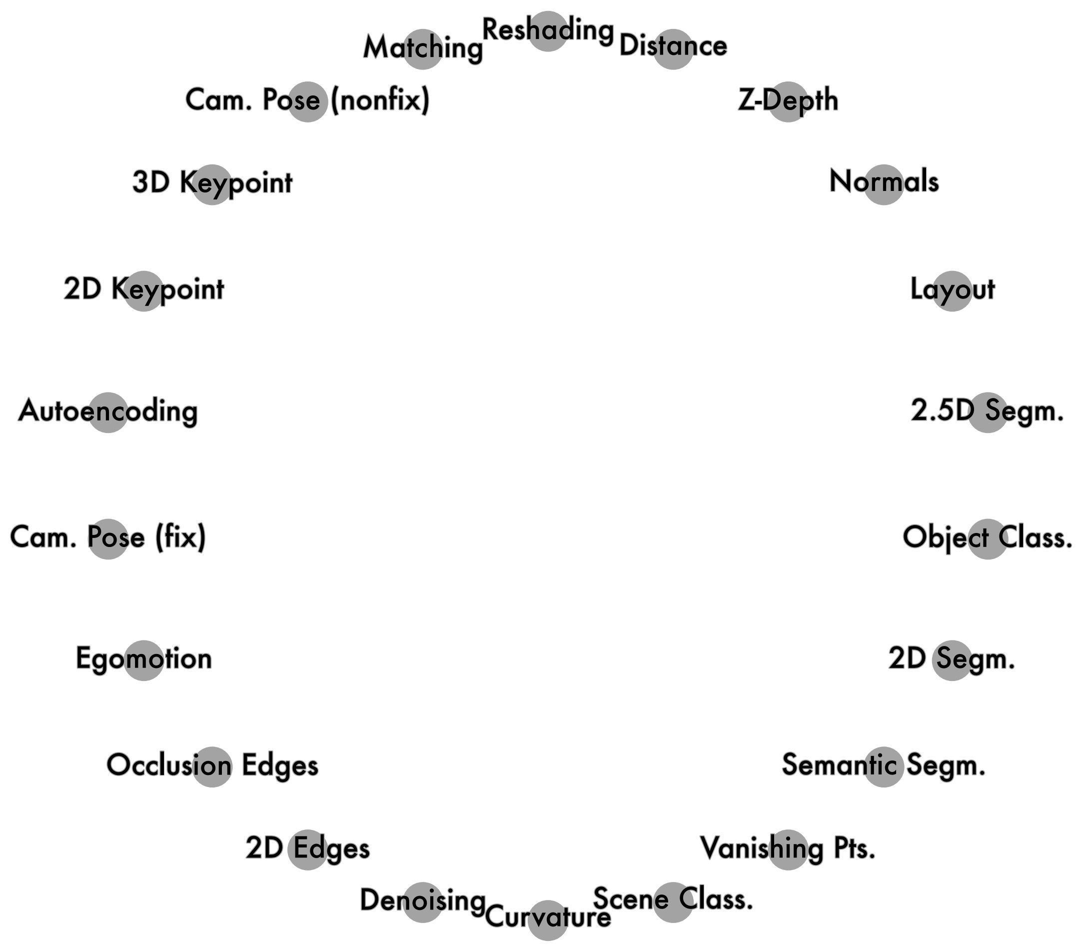


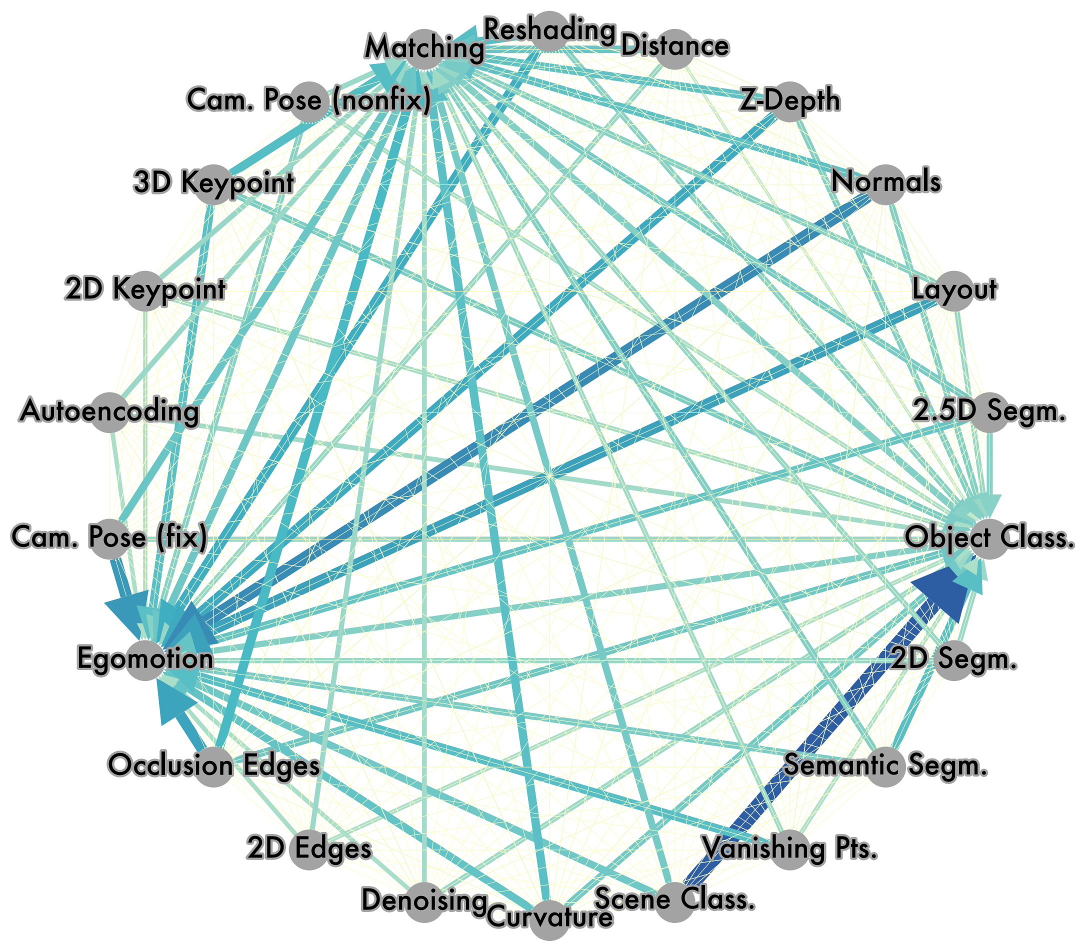
I: Task-Specific Modeling → II: Transfer Modeling → III: Normalization (AHP) → IV: Taxonomy Extraction (BIP)

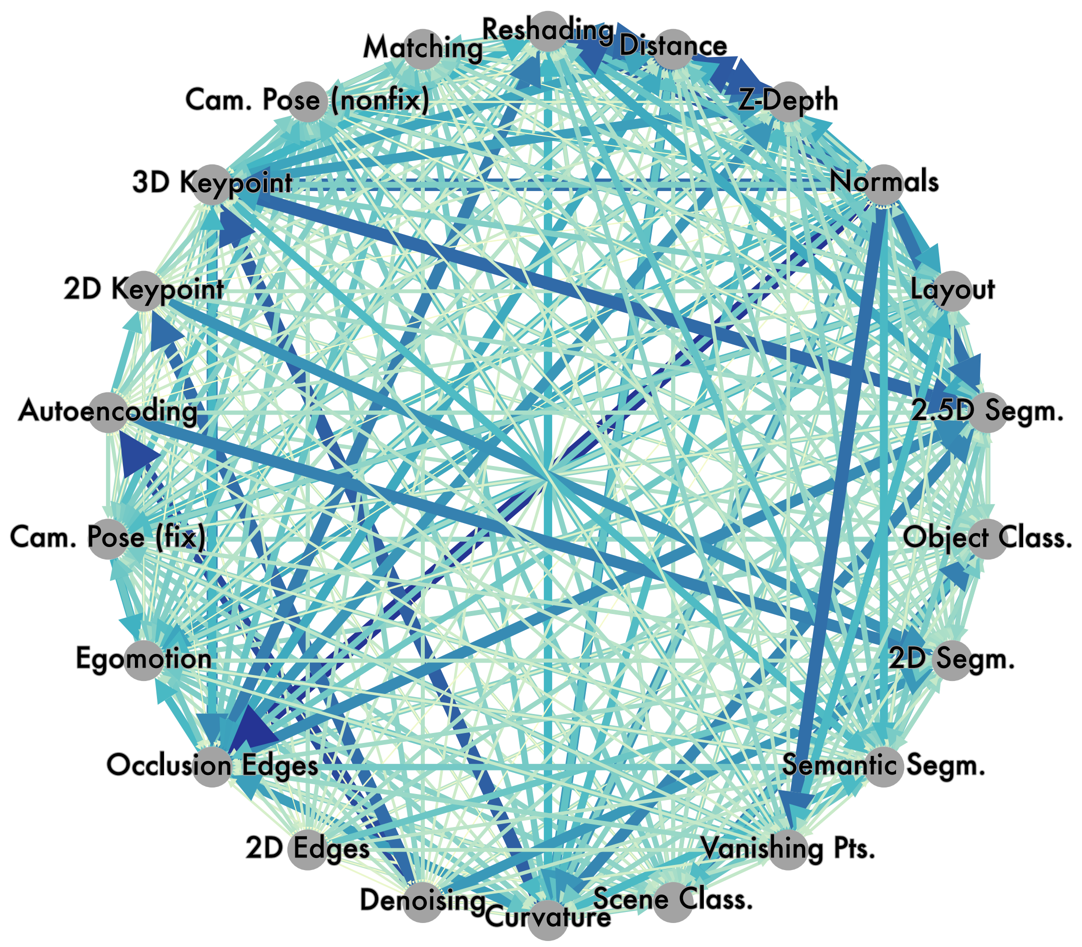
Adjacency Matrix (pre-normalization)



III: Normalization

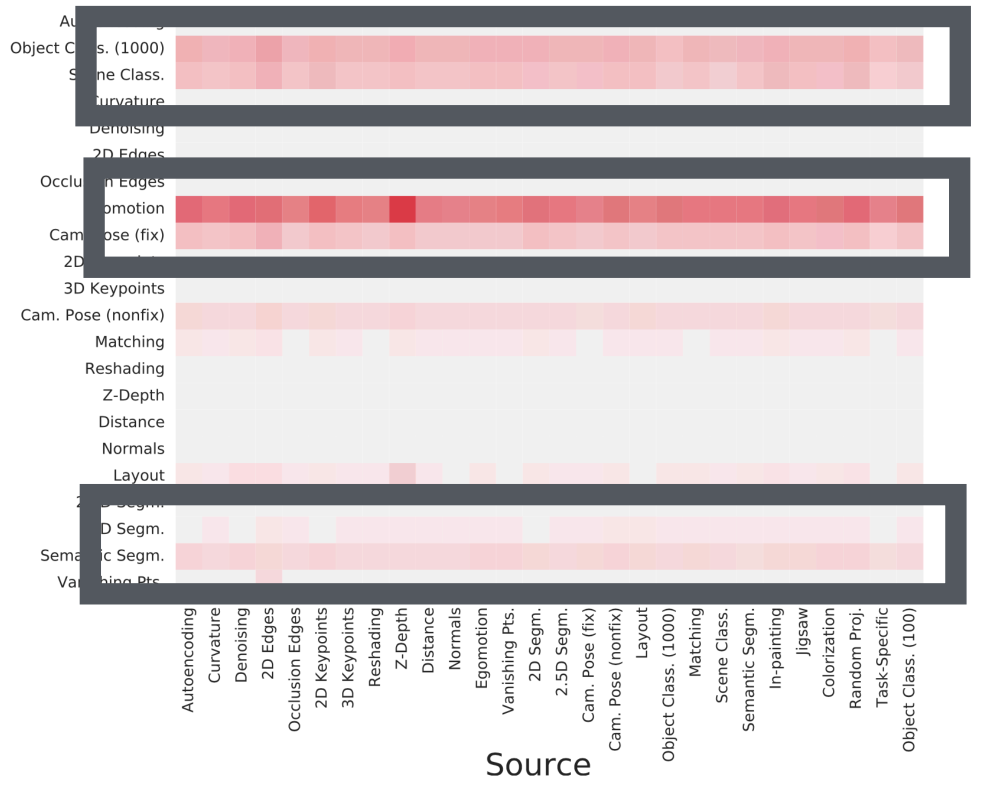




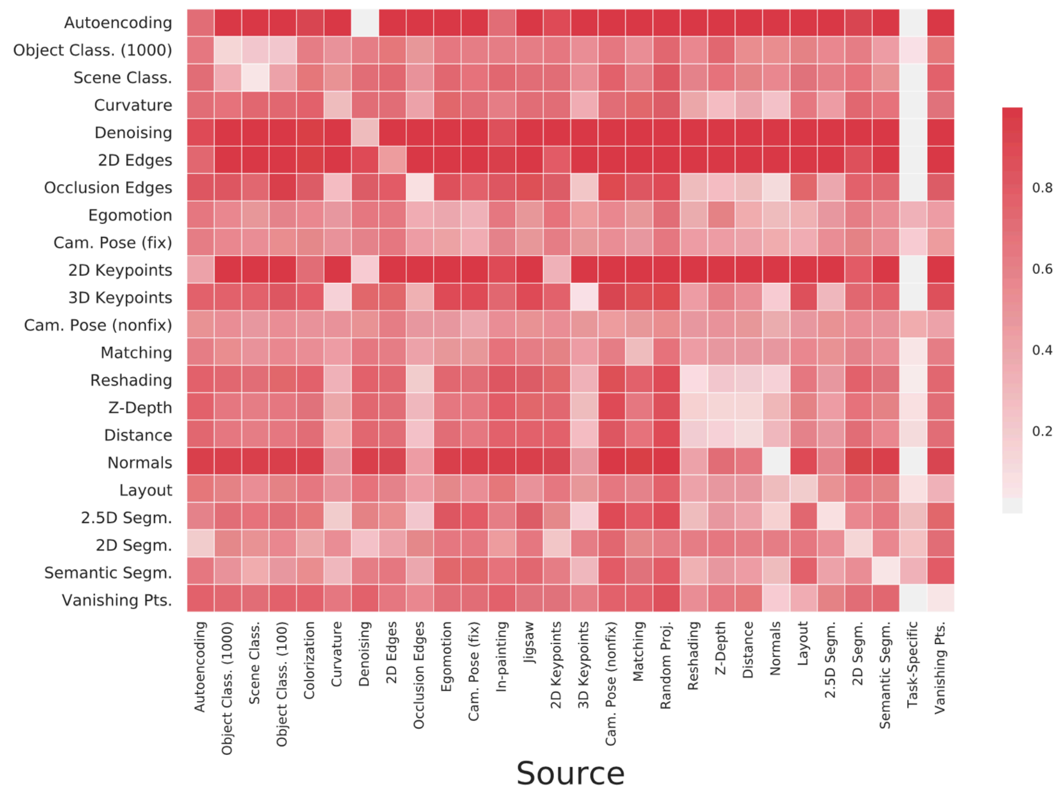


I: Task-Specific Modeling → II: Transfer Modeling → III: Normalization (AHP) → IV: Taxonomy Extraction (BIP)

Adjacency Matrix (pre-normalization)

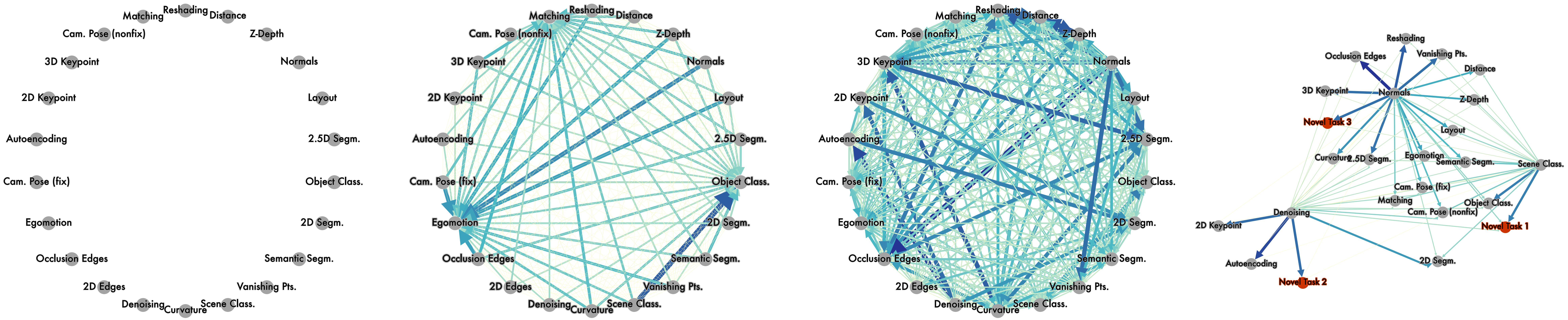


Adjacency Matrix (post-normalization)



Ordinal Normalization - Analytic Hierarchical Process. (AHP)

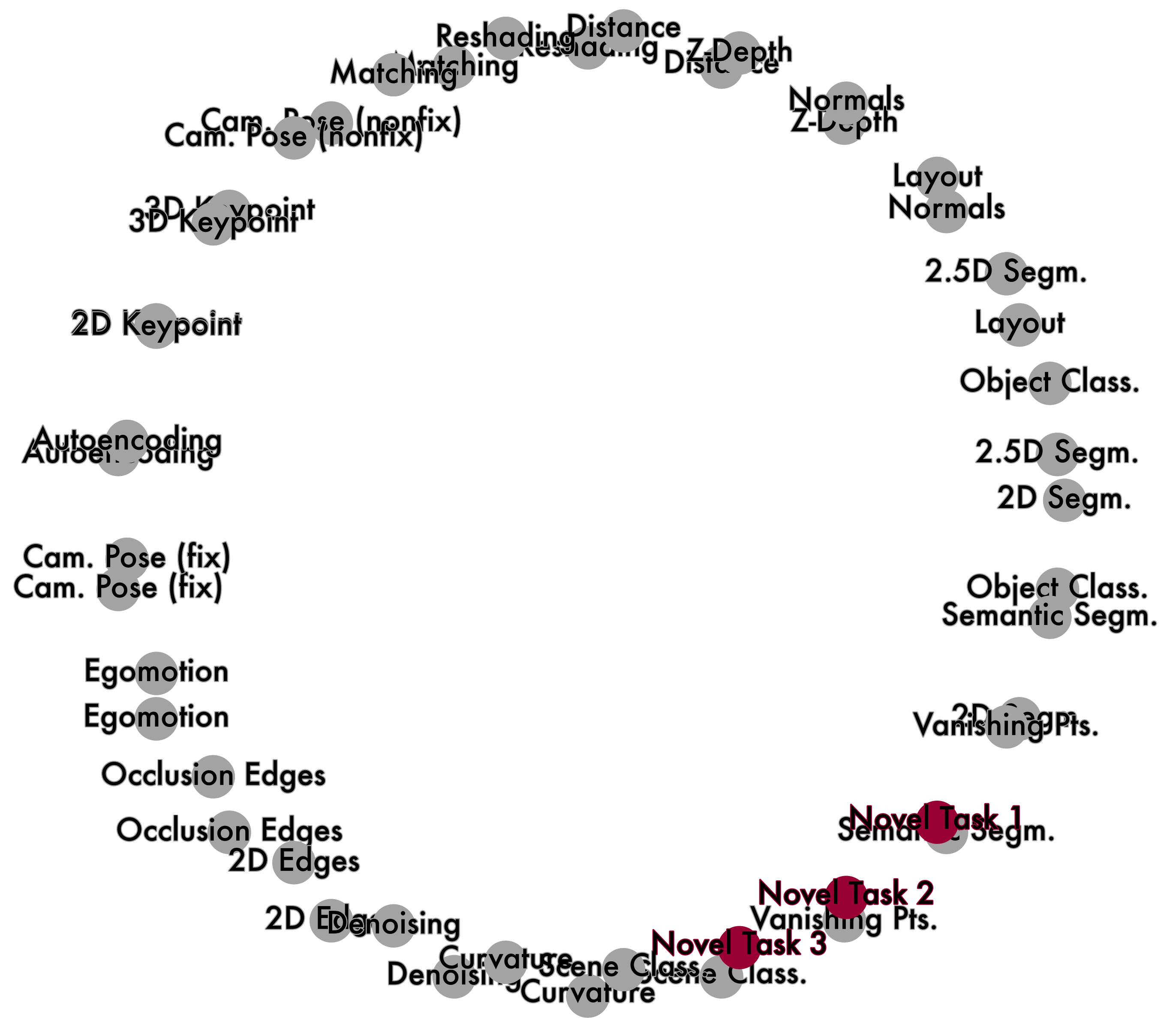
IV: Taxonomy Extraction



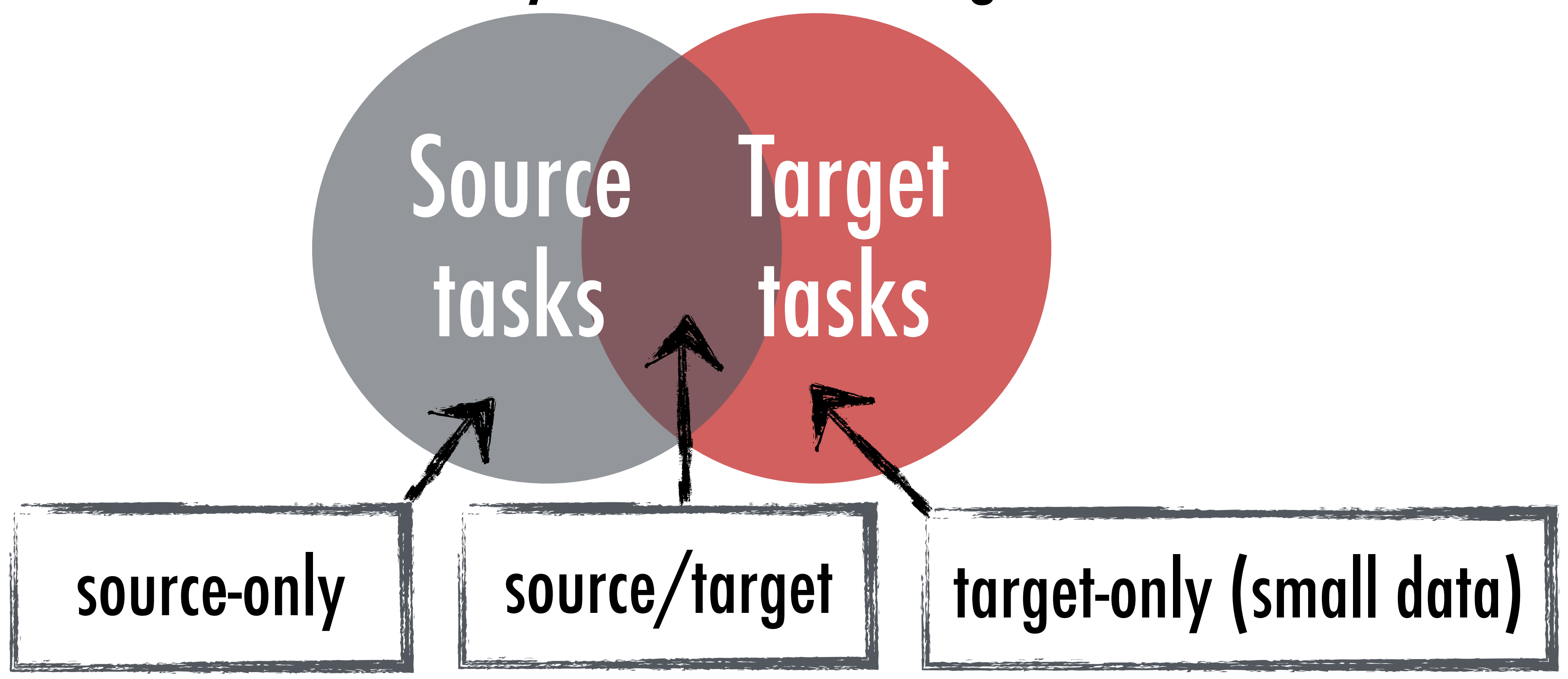
I: Task-Specific Modeling → II: Transfer Modeling → III: Normalization (AHP) → IV: Taxonomy Extraction (BIP)

- **Taxonomical structure:**
  - Sparsified
  - What are best source tasks
  - What sources for each target
  - Out-of-dictionary tasks
  - Maximize performance while constrained by some budget

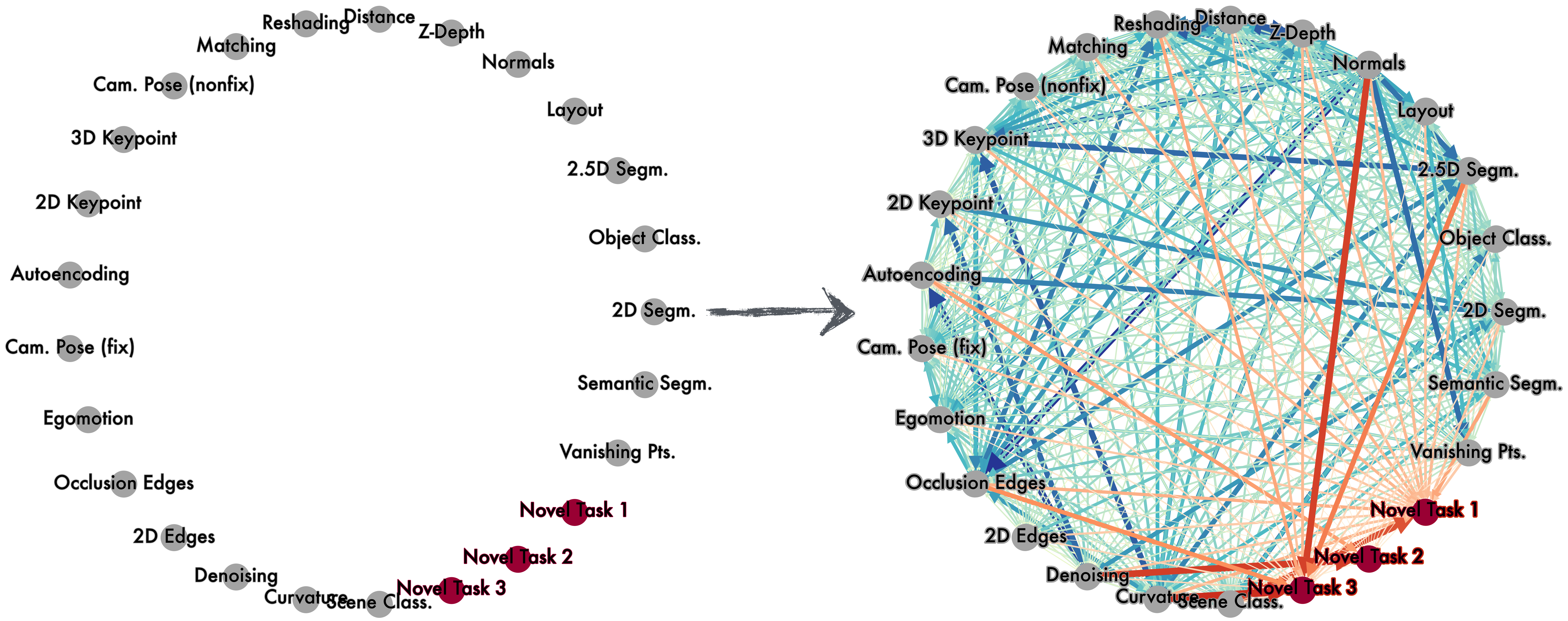
IV: Taxonomy Extraction



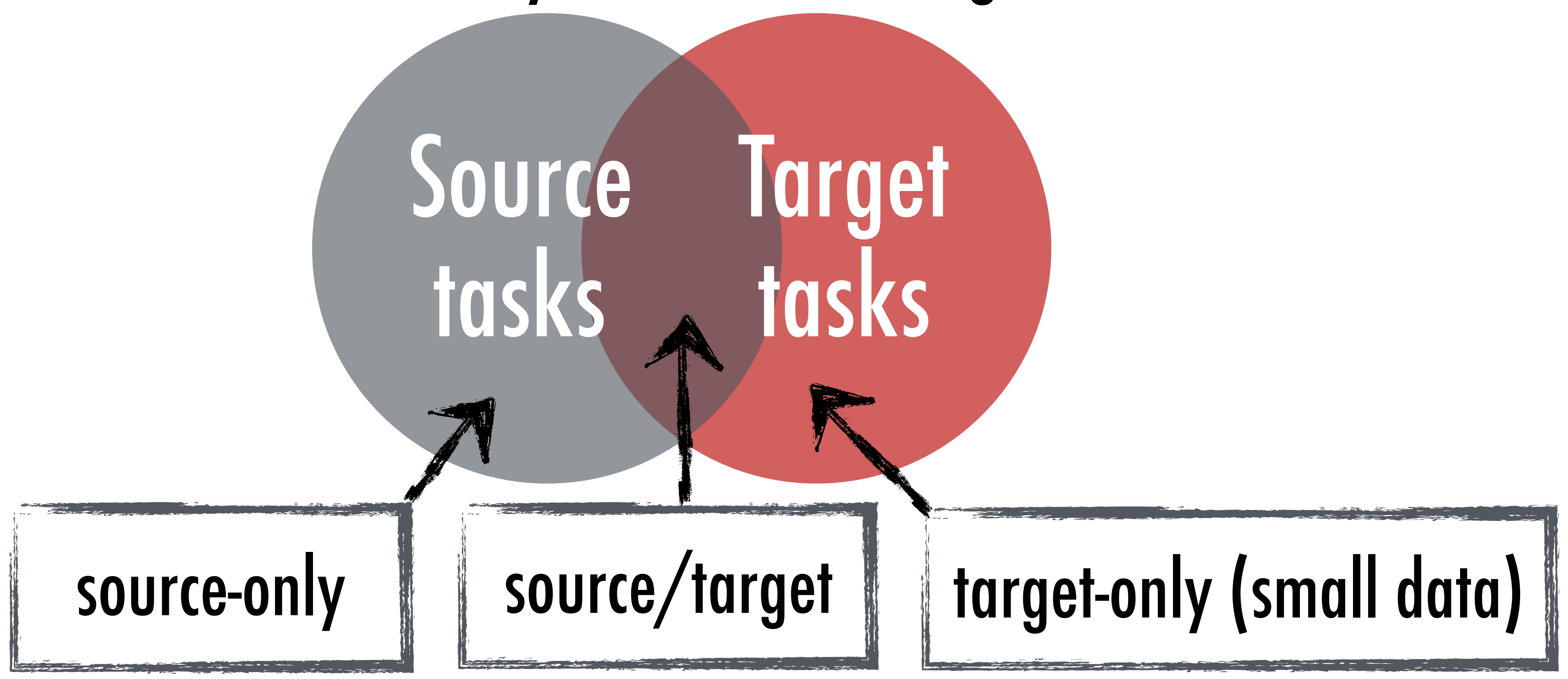
Dictionary= Sources ∪ Targets



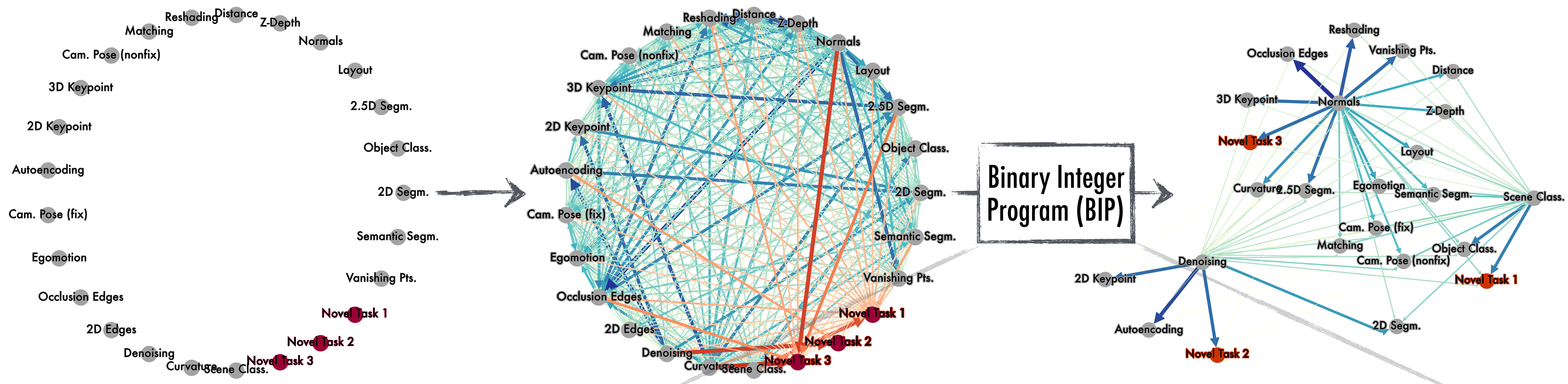
IV: Taxonomy Extraction

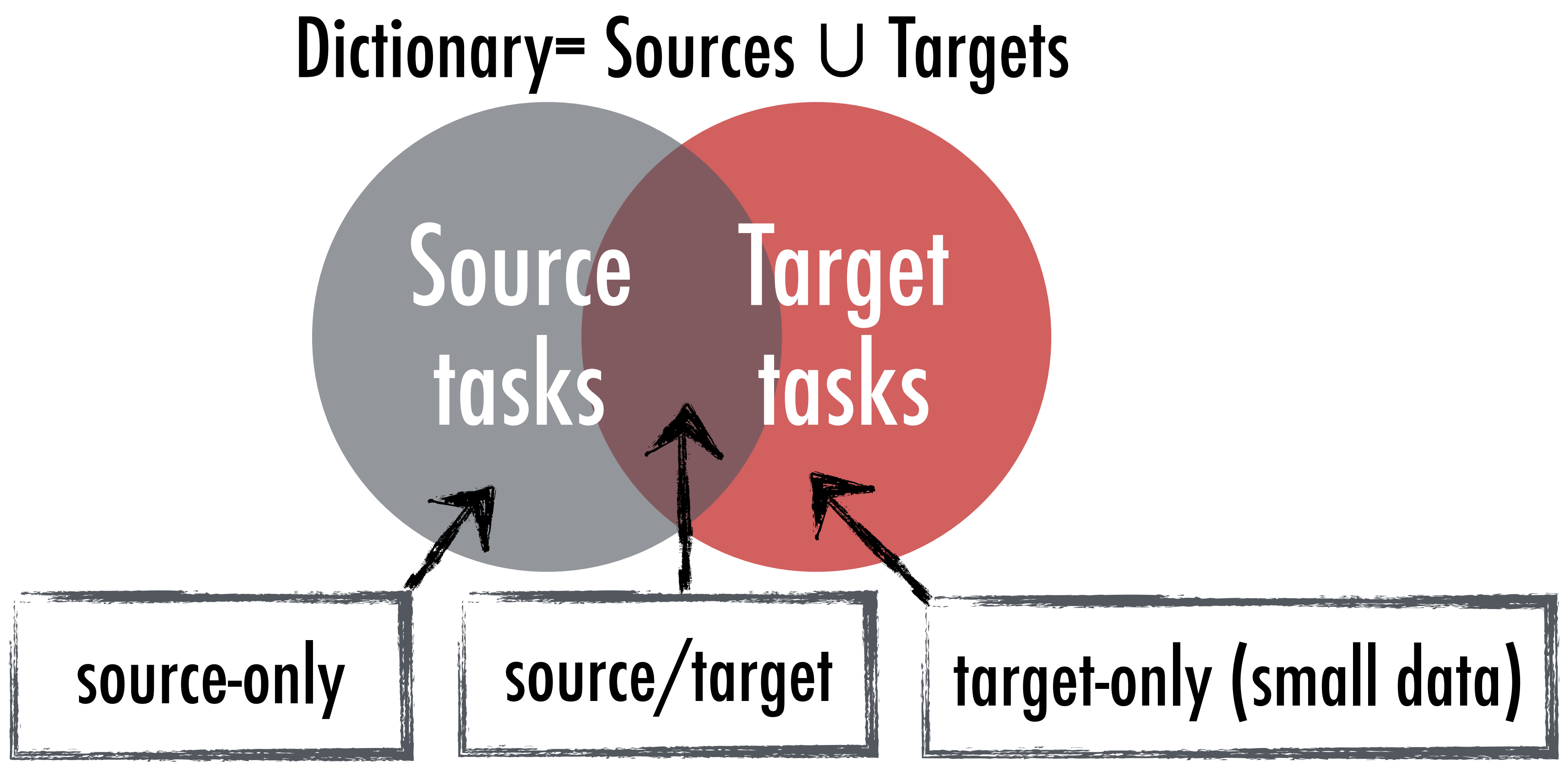


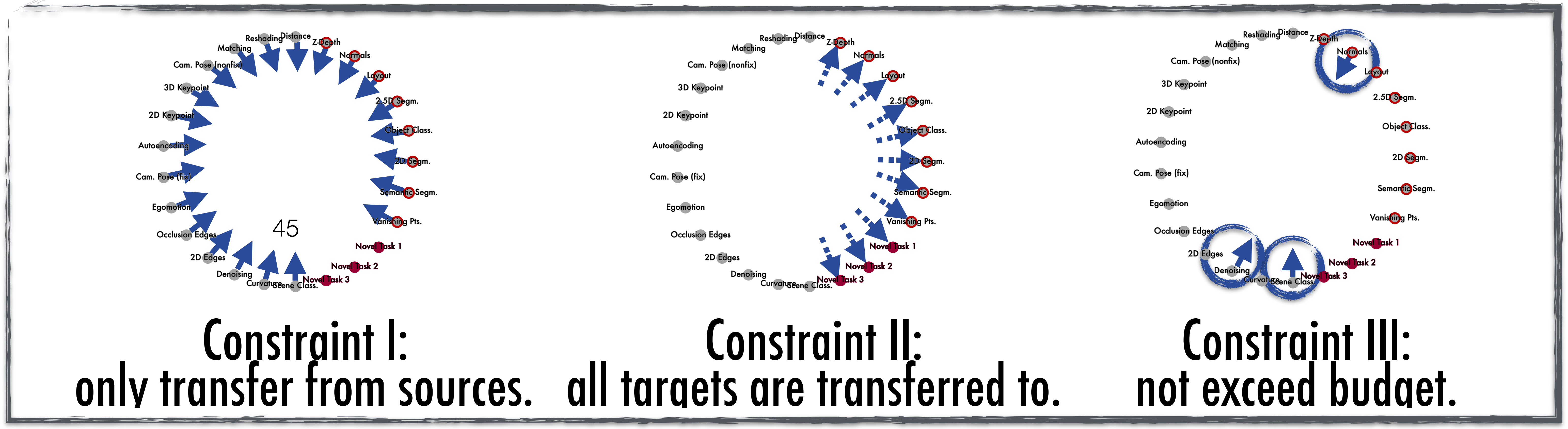
Dictionary= Sources ∪ Targets



IV: Taxonomy Extraction

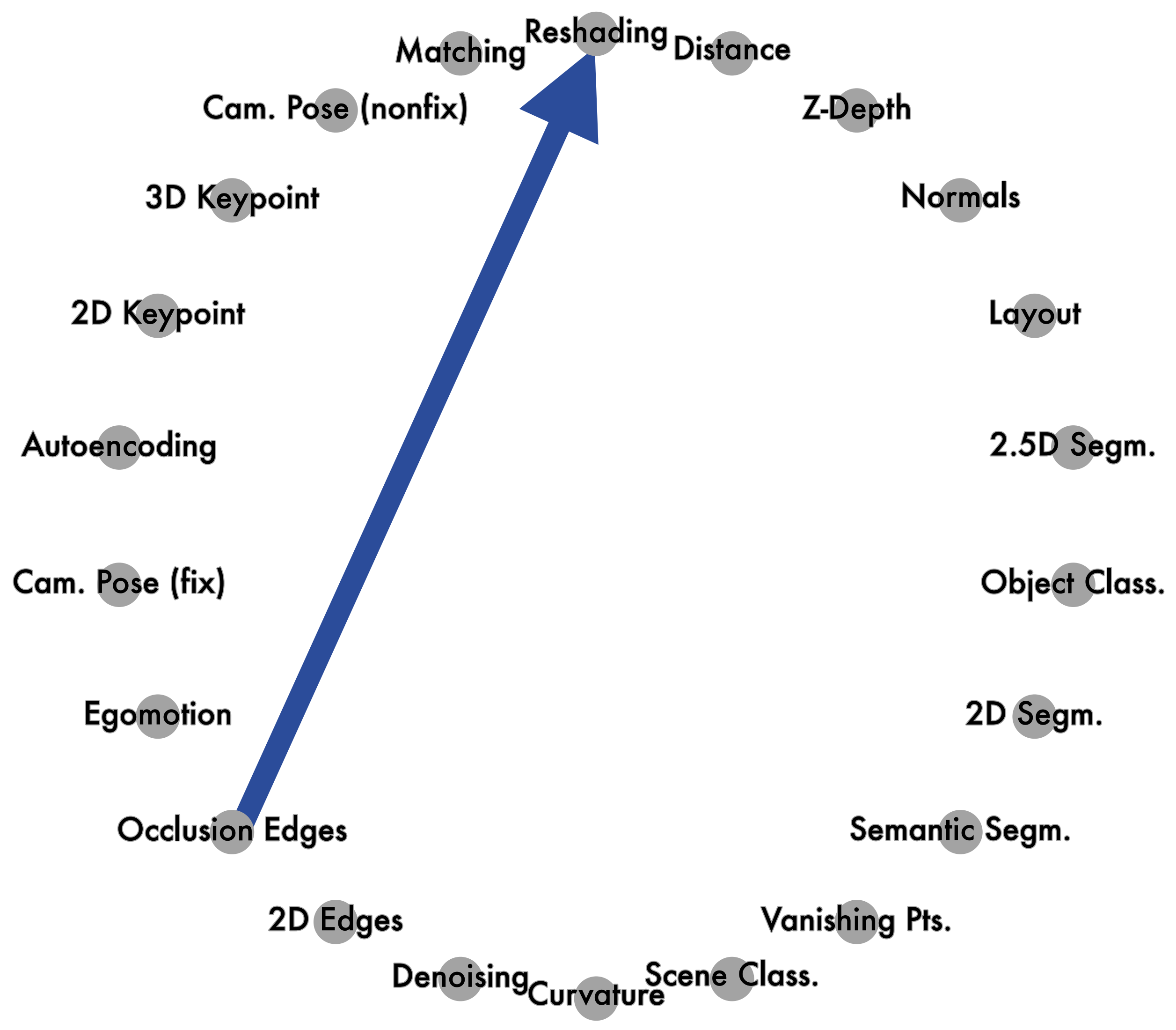




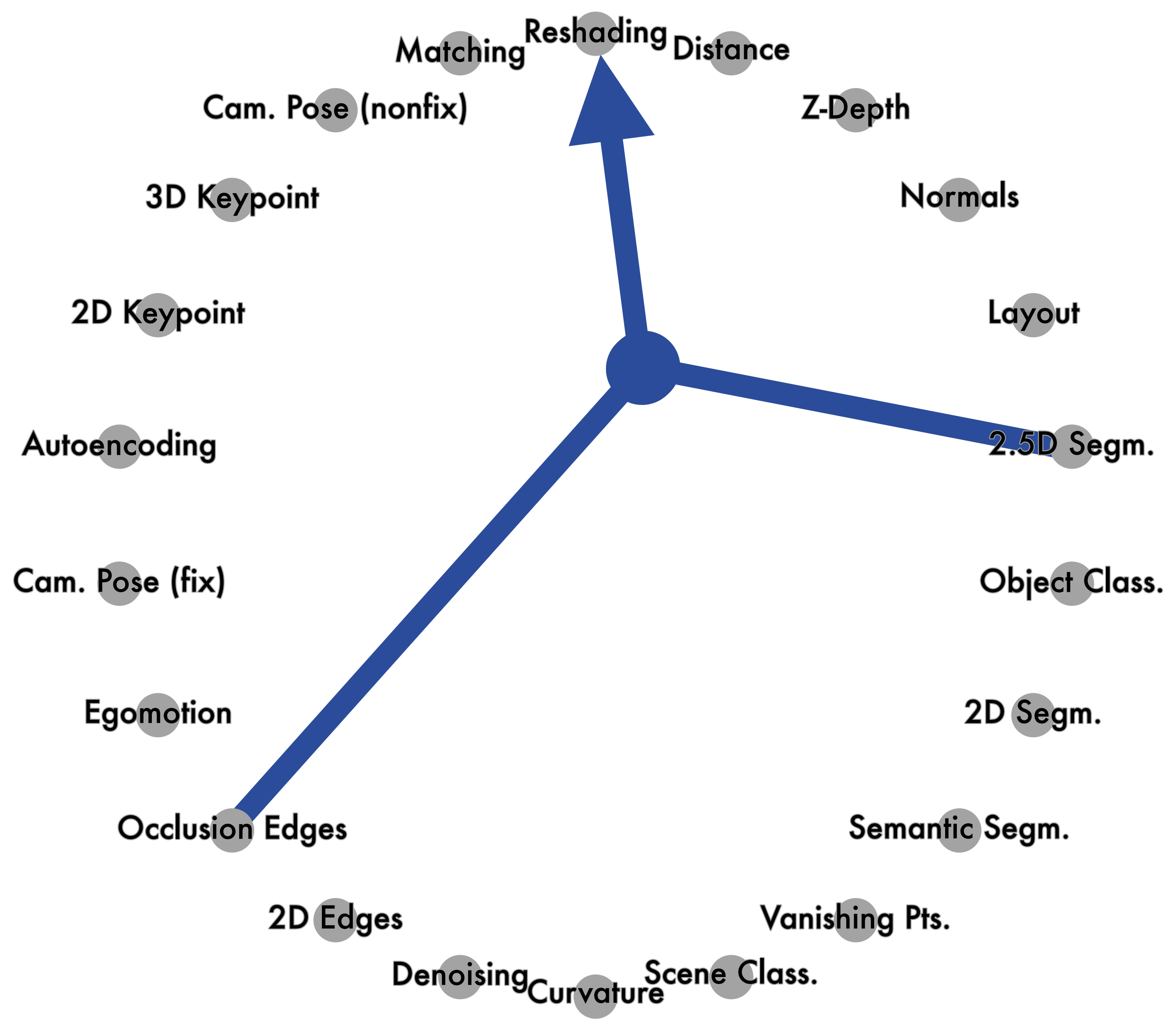


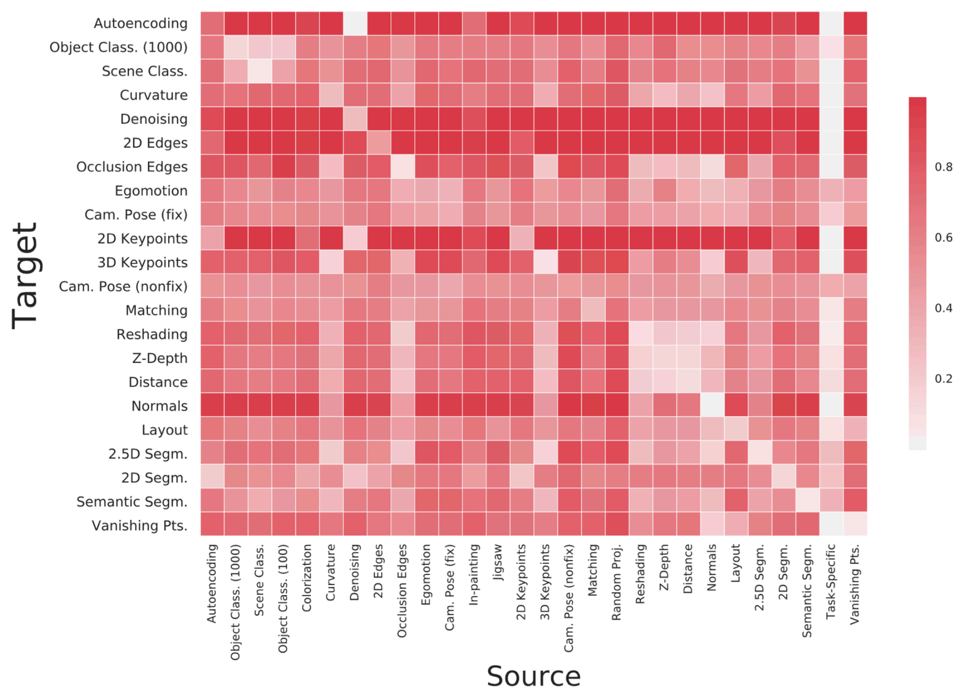
Higher Order Transfers

1st order transfer



2nd order transfer





Contents

Experimental Results

- 26 Task-Specific Networks
- 3000 Transfer Networks
- 50,000+ GPU hours

- 120k images from zero learning&16k images from transfer learning
- Transfers training data: 8x-120x less than task-specific

The main works

- the main works which Taskonomy is used do is:
  - 1.Using it to solve a group tasks
  - 2.Using its task dictionary to solve some problems with little data

How to measure effectiveness

- Gain:
  - Possibility that transfer learning is better than learning from zero
- Quality:
  - Possibility that transfer learning with a small mount of data is superior to learning from a large amount of data from zero.

Gain and Quality chart

Supervision Budget Increase (→)

Gain — max transfer order=1

Budget	2	4	6	8	12	15	22	26
2.5D Segm.	.8	.8	.8	.8	.8	.8	.8	.8
2D Edges	.0	1.	1.	1.	1.	1.	1.	1.
2D Keypt.	.0	1.	1.	1.	1.	1.	1.	1.
2D Segm.	.5	.5	.5	.5	.5	.5	.7	.7
3D Keypt.	.8	.8	.8	.9	.9	.9	.9	.9
Autoenc.	.1	.1	1.	1.	1.	1.	1.	1.
Curvature	.8	.8	.8	.8	.9	.9	.9	.9
Denoising	1.	1.	1.	1.	1.	1.	1.	1.
Distance	.8	.8	.8	.8	.9	.9	.9	.9
Egomotion	.7	.7	.7	.7	.7	.7	.7	.7
Layout	.8	.8	.8	.8	.8	.9	.9	.9
Matching	.6	.6	.6	.6	.8	.8	.8	.8
Normals	1.	1.	1.	1.	1.	1.	1.	1.
Object Cls.	.6	.6	.8	.8	.8	.8	.9	.9
Occ. Edges	.9	.9	.9	.9	.9	1.	1.	1.
Pose (fix)	.7	.7	.7	.7	.7	.8	.8	.8
Pose (nonfix)	.6	.6	.6	.6	.6	.6	.6	.6
Reshading	.9	.9	.9	.9	.9	.9	.9	.9
Scene Cls.	.7	.7	1.	1.	1.	1.	1.	1.
Sem. Segm.	.7	.7	.7	.9	.9	.9	.9	.9
Vanishing	.9	.9	.9	.9	1.	1.	1.	1.
Z-Depth	.8	.8	.8	.8	.9	.9	.9	.9

Gain — max transfer order=4

Budget	2	4	6	8	12	15	22	26
2.5D Segm.	.8	.8	.8	.8	.8	.8	.8	.8
2D Edges	.0	1.	1.	1.	1.	1.	1.	1.
2D Keypt.	.0	1.	1.	1.	1.	1.	1.	1.
2D Segm.	.5	.5	.5	.5	.5	.5	.7	.7
3D Keypt.	.8	.8	.9	.9	.9	.9	.9	.9
Autoenc.	.1	.1	.1	1.	1.	1.	1.	1.
Curvature	.8	.8	.8	.9	.9	.9	.9	.9
Denoising	1.	1.	1.	1.	1.	1.	1.	1.
Distance	.8	.8	.9	.9	.9	.9	.9	.9
Egomotion	.7	.7	.7	.7	.8	.8	.8	.8
Layout	.8	.8	.8	.8	.8	.9	.9	.9
Matching	.6	.6	.7	.7	.7	.8	.8	.8
Normals	1.	1.	1.	1.	1.	1.	1.	1.
Object Cls.	.6	.6	.8	.8	.8	.9	.9	.9
Occ. Edges	.9	.9	.9	.9	.9	.9	1.	1.
Pose (fix)	.7	.7	.7	.7	.8	.8	.8	.8
Pose (nonfix)	.6	.6	.6	.6	.6	.6	.6	.6
Reshading	.9	.9	.9	.9	.9	.9	.9	.9
Scene Cls.	.7	.7	1.	1.	1.	1.	1.	1.
Sem. Segm.	.7	.7	.8	.8	.9	.9	.9	.9
Vanishing	.9	.9	.9	.9	1.	1.	1.	1.
Z-Depth	.8	.8	.8	.8	.9	.9	.9	.9

Quality — max transfer order=1

Budget	2	4	6	8	12	15	22	26
2.5D Segm.	.3	.3	.3	.4	.4	.4	.5	.5
2D Edges	.0	.5	.5	.5	.5	.5	.5	.5
2D Keypt.	.0	.5	.5	.5	.5	.5	.5	.5
2D Segm.	.3	.3	.4	.4	.4	.4	.5	.5
3D Keypt.	.2	.2	.2	.5	.5	.5	.5	.5
Autoenc.	.0	.0	.5	.5	.5	.5	.5	.5
Curvature	.2	.2	.2	.2	.5	.5	.5	.5
Denoising	.5	.5	.5	.5	.5	.5	.5	.5
Distance	.3	.3	.3	.3	.4	.4	.5	.5
Egomotion	.5	.5	.5	.5	.5	.5	.5	.5
Layout	.3	.3	.3	.3	.3	.5	.5	.5
Matching	.2	.2	.2	.2	.5	.5	.5	.5
Normals	.5	.5	.5	.5	.5	.5	.5	.5
Object Cls.	.2	.2	.4	.4	.4	.4	.5	.5
Occ. Edges	.3	.3	.3	.3	.3	.5	.5	.5
Pose (fix)	.4	.4	.4	.4	.4	.5	.5	.5
Pose (nonfix)	.5	.5	.5	.5	.5	.5	.5	.5
Reshading	.4	.4	.4	.4	.4	.4	.5	.5
Scene Cls.	.1	.1	.5	.5	.5	.5	.5	.5
Sem. Segm.	.2	.2	.2	.5	.5	.5	.5	.5
Vanishing	.2	.2	.2	.2	.5	.5	.5	.5
Z-Depth	.3	.3	.3	.3	.5	.5	.5	.5

Quality — max transfer order=4

Budget	2	4	6	8	12	15	22	26
2.5D Segm.	.3	.3	.4	.4	.4	.4	.5	.5
2D Edges	.0	.5	.5	.5	.5	.5	.5	.5
2D Keypt.	.0	.5	.5	.5	.5	.5	.5	.5
2D Segm.	.3	.4	.4	.4	.4	.4	.5	.5
3D Keypt.	.2	.2	.5	.5	.5	.5	.5	.5
Autoenc.	.0	.0	.0	.5	.5	.5	.5	.5
Curvature	.2	.2	.2	.5	.5	.5	.5	.5
Denoising	.5	.5	.5	.5	.5	.5	.5	.5
Distance	.3	.3	.4	.4	.5	.5	.5	.5
Egomotion	.5	.5	.5	.5	.6	.6	.6	.6
Layout	.3	.3	.3	.3	.4	.5	.5	.5
Matching	.2	.2	.4	.4	.4	.5	.5	.5
Normals	.5	.5	.5	.5	.5	.5	.5	.5
Object Cls.	.2	.2	.4	.4	.4	.5	.5	.5
Occ. Edges	.3	.3	.4	.4	.4	.4	.5	.5
Pose (fix)	.4	.4	.4	.4	.5	.5	.5	.5
Pose (nonfix)	.5	.5	.5	.5	.5	.5	.5	.5
Reshading	.4	.4	.4	.4	.5	.5	.5	.5
Scene Cls.	.1	.1	.5	.5	.5	.5	.5	.5
Sem. Segm.	.2	.2	.2	.2	.5	.5	.5	.5
Vanishing	.2	.2	.2	.2	.5	.5	.5	.5
Z-Depth	.3	.3	.4	.4	.4	.4	.5	.5

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.

Solve new task

We can consider the target task in our dictionary as a new task, simulating the situation with only a small amount of data.

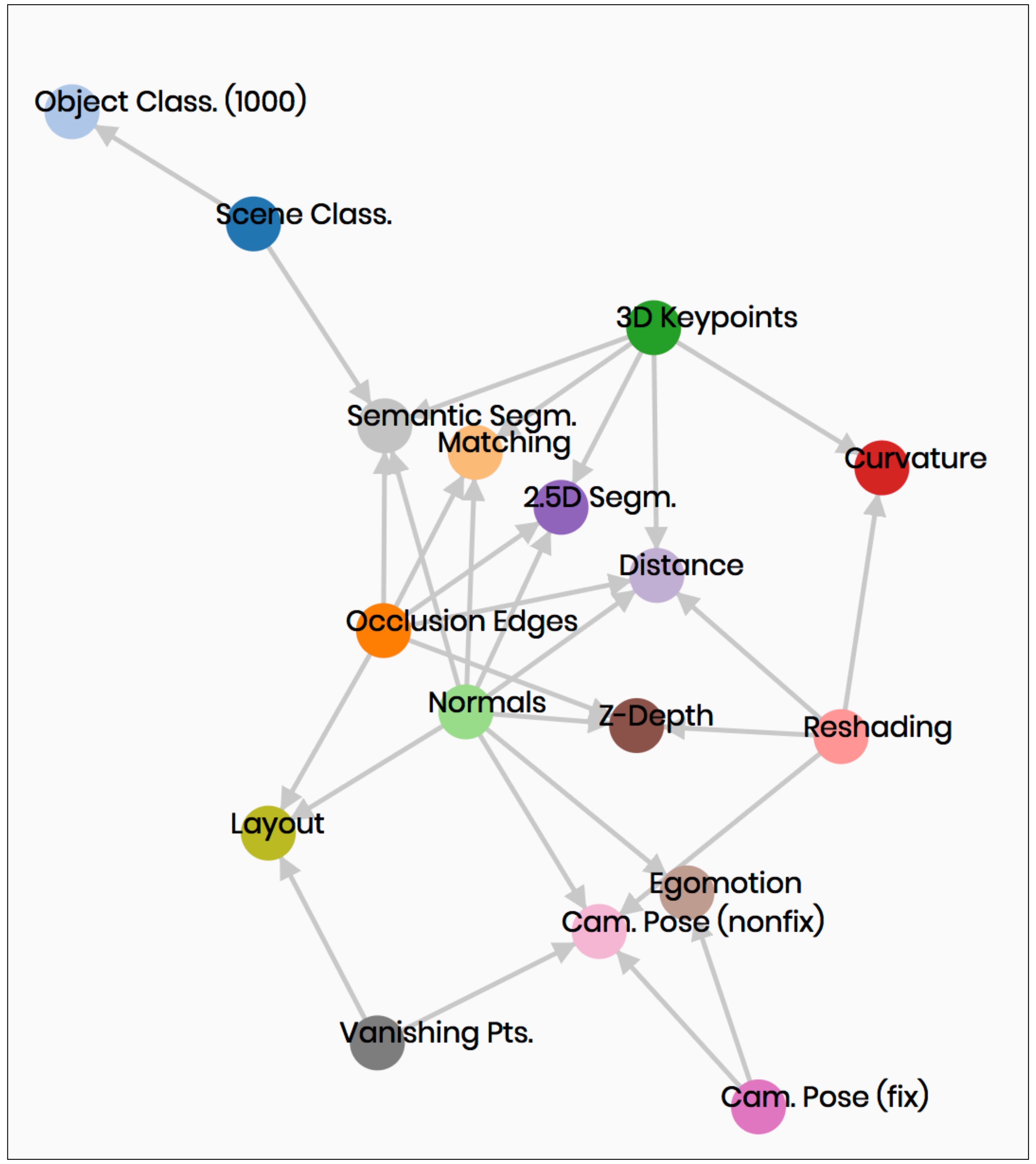
Order Increase (→)

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.

A Taxonomy

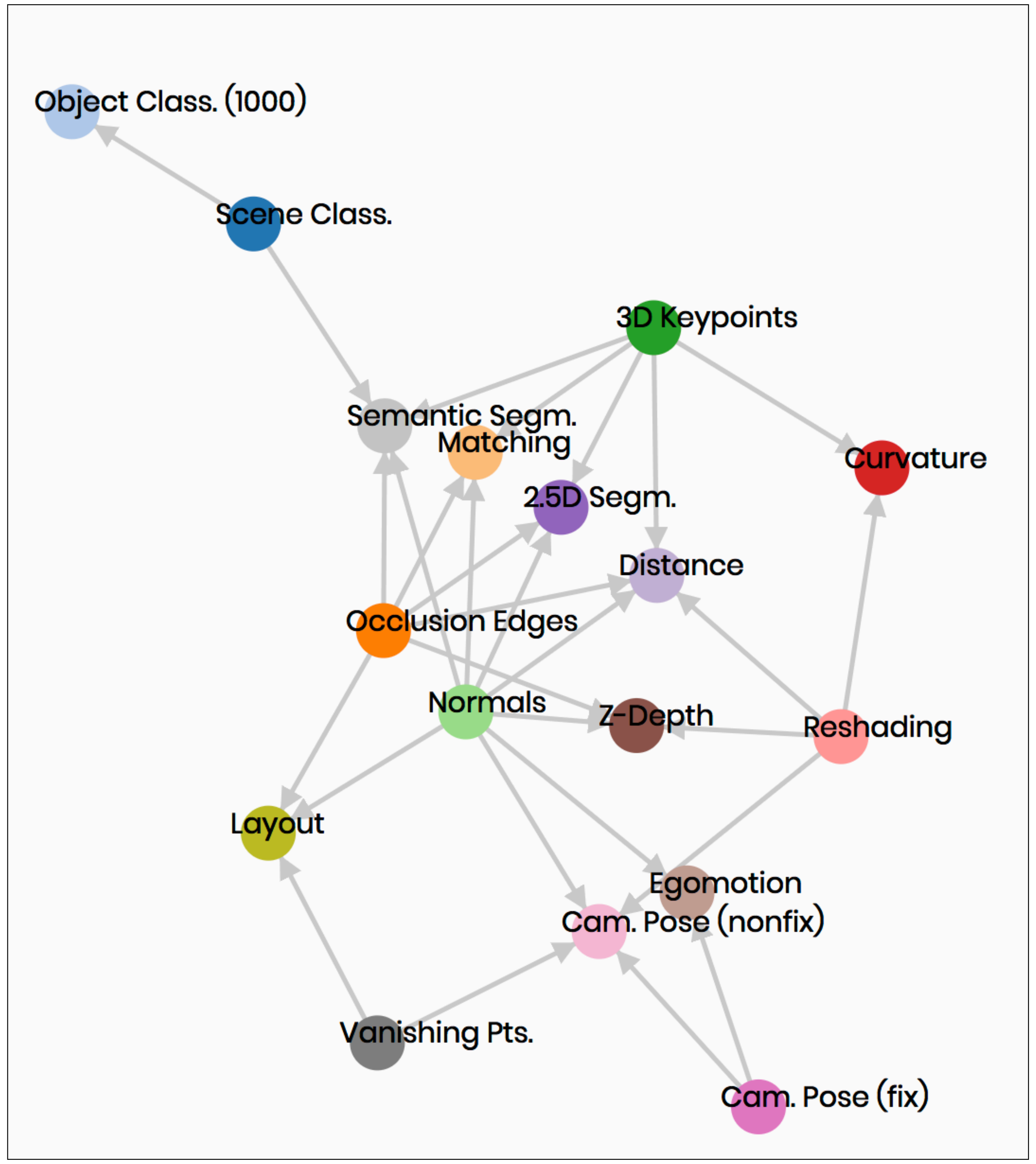


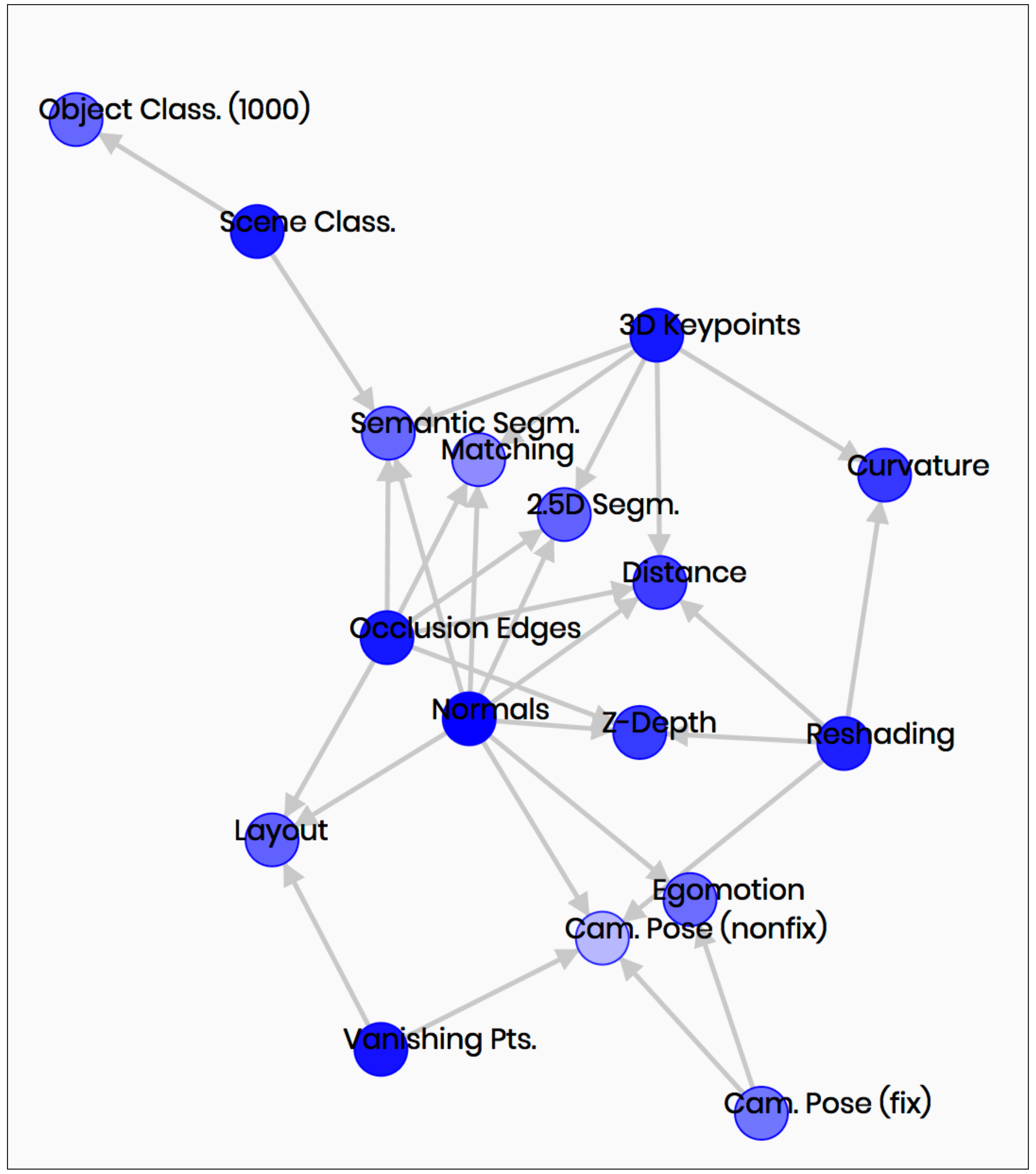
Target-only: Semantic Segmentation, Layout, Egomotion, 3D Curvature.

Source/Target: Depth, Occlusion Edges, Scene Classification, Normals, Camera Pose (non-fix), 3D Keypoint, Object Classification, Z-depth, Reshading, 3D Segmentation, Vanishing-Point, Camera pose (fix), Matching.

Source-only: Jigsaw, Colorization, Autoencoding, 2D edges, 2D edges, 2D keypoints, Denoising.

A Taxonomy



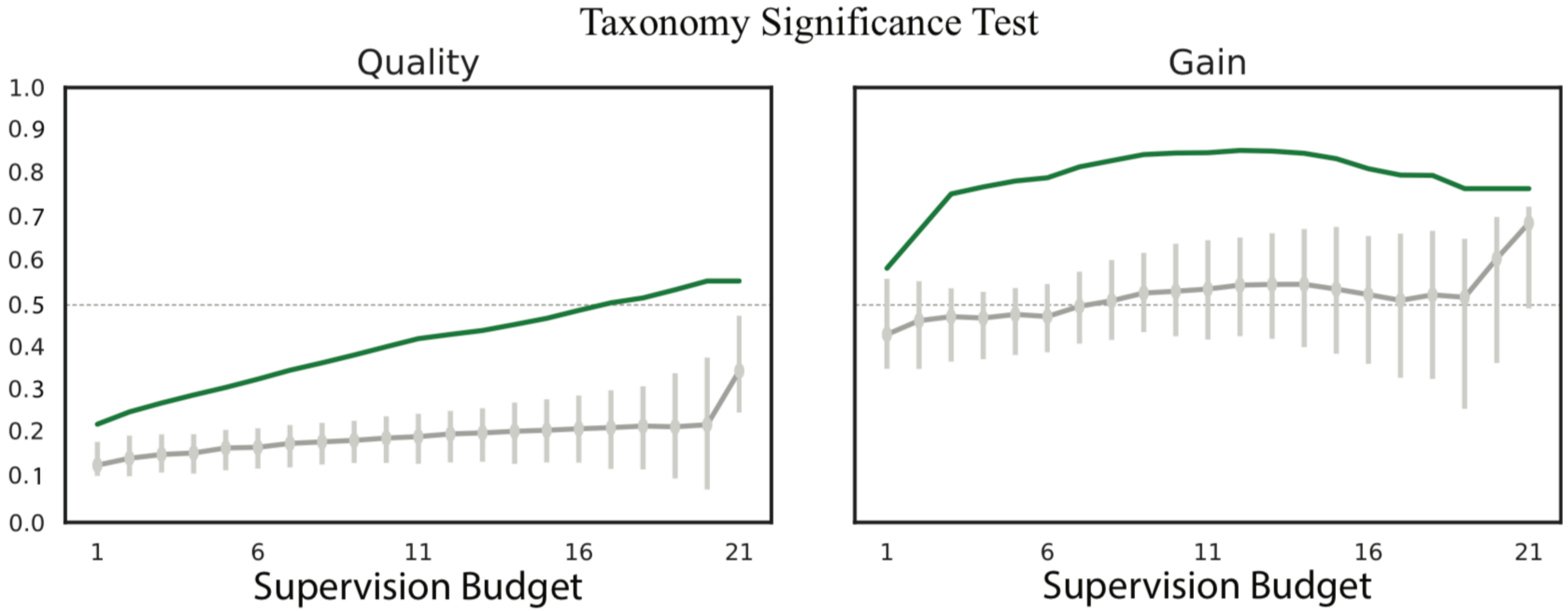


Gain: how much gained by transferring. (Win rate against a network trained from scratch using transfer data)

Experiments in paper/poster

Experiments in paper/poster

- Significance Test



Summary

- Focus on a set of tasks and use the correlation between tasks to reduce overall data usage.
- Treat tasks in concert, coming from a structured space, rather than isolated concepts.
- A striving step towards understanding the space of vision tasks.
- Solve a set of tasks pretty well with less data.

Thank you!