Taskology: Utilizing Task Relations at Scale

Yao Lu\textsuperscript{1,2}, Soren Pirk\textsuperscript{1,2}, Jan Dlabal\textsuperscript{3}, Anthony Brohan\textsuperscript{1,2}, Ankita Pasad\textsuperscript{4}, Zhao Chen\textsuperscript{5}, Vincent Casser\textsuperscript{1,6}, Anelia Angelova\textsuperscript{1,2}, Ariel Gordon\textsuperscript{1,2}

\textsuperscript{1}Robotics at Google, \textsuperscript{2}Google Research, \textsuperscript{3}Toyota Technology Institute at Chicago \textsuperscript{4}Waymo LLC

In a nutshell

- We present a method for multitask learning at scale.
- Task models supervise each other through task relations, improving each other’s performance.
- We benefit from unlabeled or partially labeled data.
- We train distributedly and asynchronously: tasks can tolerate very stale predictions from their peers.

Method

- Task relations are represented as a consistency constraint, enforced by a differentiable loss term ($\mathcal{L}^{\text{con}}$), on unlabeled data.
- Tasks may also receive direct supervision from labeled data ($\mathcal{L}^{\text{labeled}}$).
- Each task trains on a separate machine. They communicate through $\mathcal{L}^{\text{con}}$ only.

$$
\mathcal{L} = \sum_{i=1}^{n} \mathcal{L}^{\text{sup}}(\hat{y}_i(w_i, x), y_i(x)) + \mathcal{L}^{\text{con}}(\hat{y}_1(w_1, x), \hat{y}_2(w_2, x), \ldots, \hat{y}_n(w_n, x))
$$

Consistency improves the tasks’ performance:

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Depth Error (Abs. Rel.)</th>
<th>Segmentation MDU</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. Depth &amp; motion only</td>
<td>0.165</td>
<td>0.455</td>
</tr>
<tr>
<td>B. Segmentation only</td>
<td>0.129</td>
<td>-</td>
</tr>
<tr>
<td>C. Frozen segmentation model B with depth &amp; motion</td>
<td>0.129</td>
<td>0.471</td>
</tr>
<tr>
<td>D. Frozen depth &amp; motion model C &amp; segmentation</td>
<td>0.125</td>
<td>0.479</td>
</tr>
<tr>
<td>E. Depth, motion and segmentation training jointly</td>
<td>0.125</td>
<td>0.479</td>
</tr>
</tbody>
</table>

Datasets used: COCO for supervision, Cityscapes without the labels for consistency.

Example 1: Depth, motion & segmentation

These tasks are related through projective geometry, forming a differentiable consistency constraint:

- Segmenting moving objects allows decomposing motion to object motion & camera motion.
- With depth & motion, optical flow can be obtained & used to assert mask consistency & photometric consistency across frames.

Example 2: Detection & tracking in point clouds

A pair of LIDAR frames (Waymo Open Dataset)

Example 2 cont’d:

- Imposing consistency of tracking & detection improves test metrics. The less labels we used, the greater was the improvement provided by consistency.

Scalability of our method

- Parallelizable: Each task trains on a separate machine.
- Asynchronous: Each tasks sees stale predictions of its peers. Predictions as old as 2000 training steps did not hurt the accuracy.
- Agnostic to the internals of the tasks’ models. If it can output predictions and receive gradients, it’s a go.
- Easy to add tasks: Each model trains on its own hardware, with its favorite hyperparams, as published by its author.

Summary

- Main contribution: modular design for multitask training.
- Task relations are utilized through consistency losses.
- Unlabeled and simulated data can be used to improve performance in the underlabeled regime.
- Distributed, robust, asynchronous, scalable training algorithm.
- Future direction: Can learned differentiable constraints be used similarly?