Taskology: Utilizing Task Relations at Scale

Yao Lu[1,2], Soren Pirk[1,2], Jan Dlabal[2], Anthony Brohan[1,2], Ankita Pasad[3], Zhao Chen[4], Vincent Casser[4], Anelia Angelova[1,2], Ariel Gordon[1,2]

[1]Robotics at Google, [2]Google Research, [3]Toyota Technology Institute at Chicago [4]Waymo LLC

In a nutshell

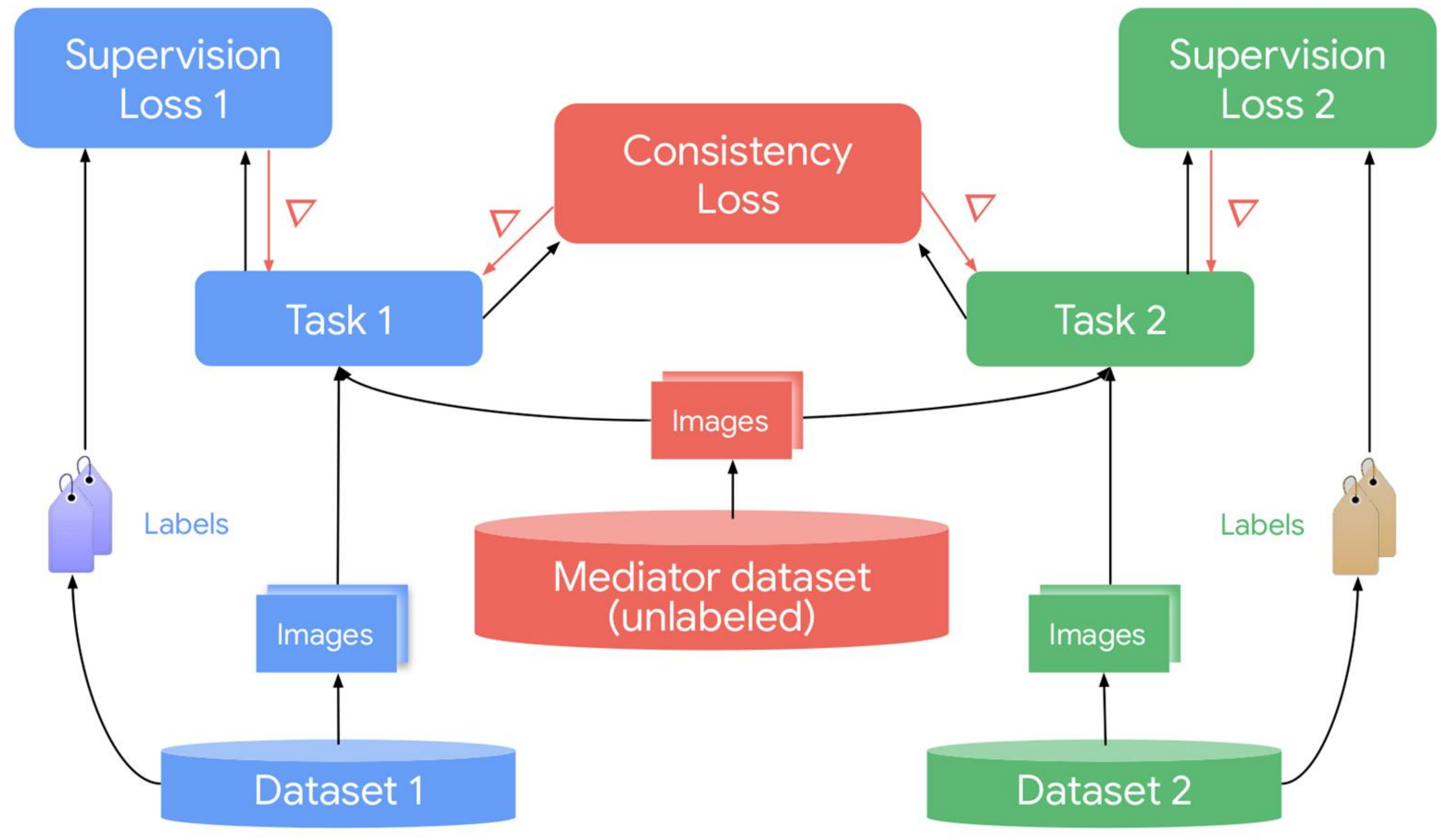
- We present a method for multitask learning at scale.
- Task models supervise each other through task relations, improving each other's performance.
- We benefit from unlabeled or partially labeled data.
- We train distributedly and asynchronously: tasks can tolerate very stale predictions from their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

Method

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

Loss

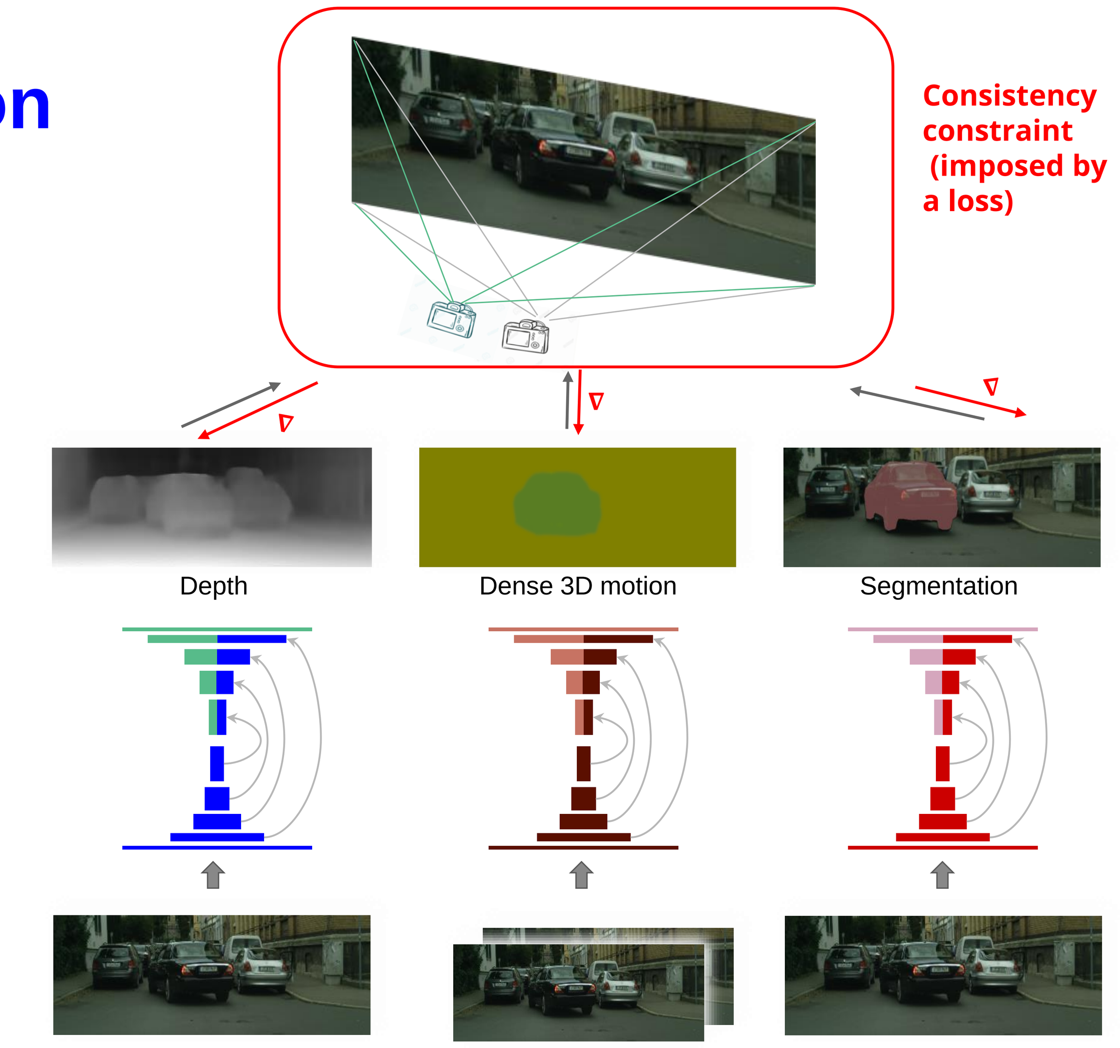
$$\mathcal{L} = \sum_{i=1}^{n} \mathcal{L}_i^{sup}\Big(\hat{y}_i(w_i, x), y_i(x)\Big) + \mathcal{L}^{con}\Big(\hat{y}_1(w_1, x), \hat{y}_2(w_2, x), \ldots, \hat{y}_n(w_n, x)\Big)$$



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.

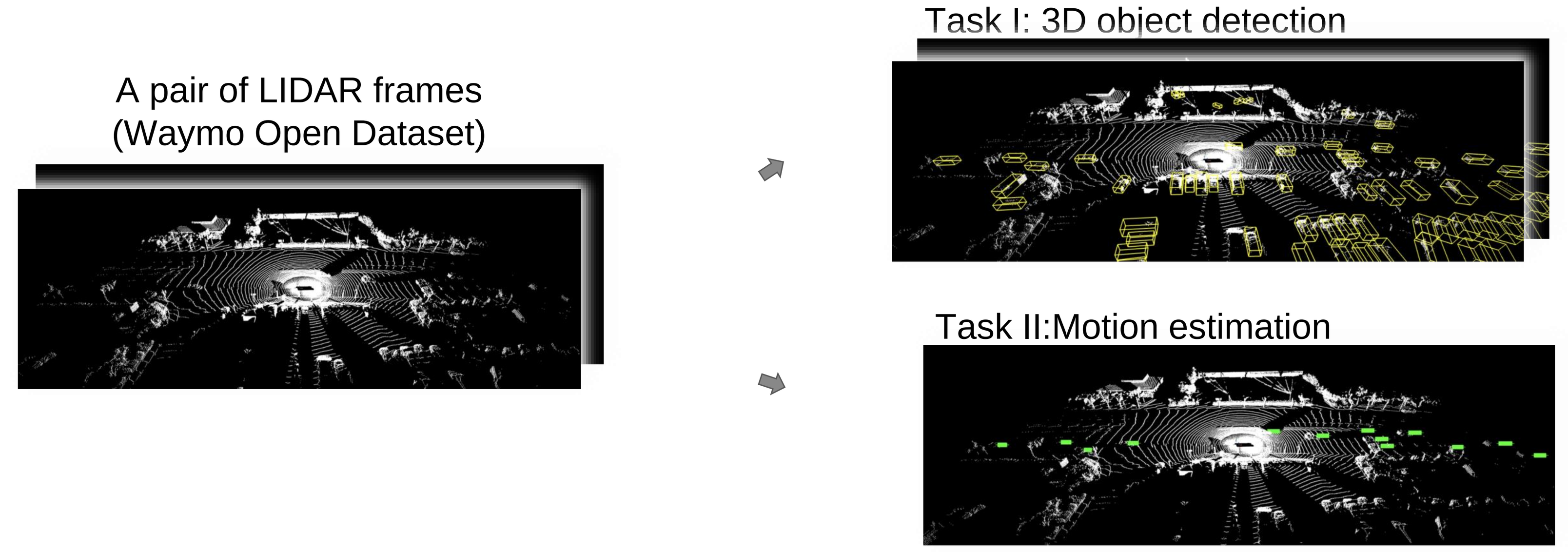


Consistency improves the tasks' performance:

Configuration	Depth Error (Abs. Rel.)	Segmentation MIOU
A. Depth & motion only	0.165	-
B. Segmentation only	-	0.455
C. Frozen segmentation model B with depth & motion	0.129	-
D. Frozen depth & motion model C & segmentation	-	0.471
E. Depth, motion and segmentation training jointly	**0.125**	**0.478**

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

Example 2: Detection & tracking in point clouds



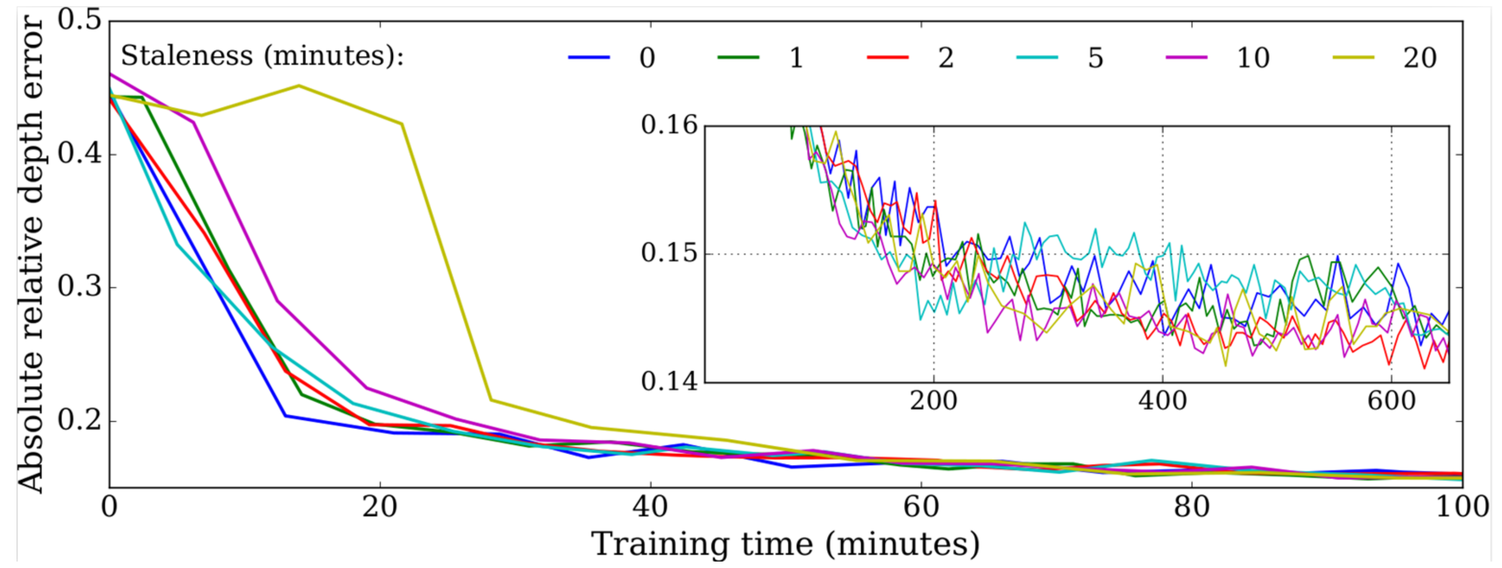
Example 2 cont'd:

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

Method	Labels	3D mAP/mAPH (%)
No Consistency	5%	17.6/9.6
Adding $\mathcal{L}^{con}$	5%	**23.5/12.0**
No Consistency	20%	30.8/16.4
Adding $\mathcal{L}^{con}$	20%	**31.6/19.1**
No Consistency	100%	53.0/47.6
Adding $\mathcal{L}^{con}$	100%	**54.2/49.6**

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.



Effect of peer model staleness on the training of depth & egomotion: using peer predictions as stale as 2000 training steps old (20 minutes) resulted in no adverse effect on the test metrics.

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?