Learning Multiple Dense Prediction Tasks from Partially Annotated Data

Citation for published version:
Li, W, Bilen, H & Liu, X 2022, Learning Multiple Dense Prediction Tasks from Partially Annotated Data. in *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Institute of Electrical and Electronics Engineers (IEEE), pp. 18857-18867, IEEE/CVF Conference on Computer Vision and Pattern Recognition 2022
, New Orleans, Louisiana, United States, 19/06/22. https://doi.org/10.1109/CVPR52688.2022.01831

Digital Object Identifier (DOI):
10.1109/CVPR52688.2022.01831

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

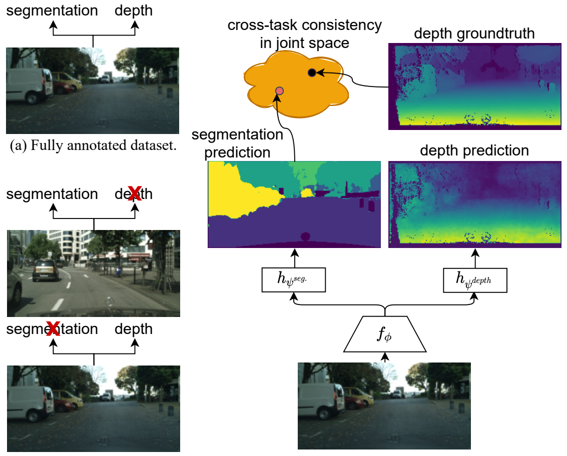
Published In:
Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

Learning Multiple Dense Prediction Tasks from Partially Annotated Data

Wei-Hong Li, Xialei Liu, and Hakan Bilen
VICO Group, University of Edinburgh, United Kingdom
github.com/VICO-UoE/MTPSL

Abstract

*Despite the recent advances in multi-task learning of dense prediction problems, most methods rely on expensive labelled datasets. In this paper, we present a label efficient approach and look at jointly learning of multiple dense prediction tasks on partially annotated data (*i.e. *not all the task labels are available for each image), which we call multi-task partially-supervised learning. We propose a multi-task training procedure that successfully leverages task relations to supervise its multi-task learning when data is partially annotated. In particular, we learn to map each task pair to a joint pairwise task-space which enables sharing information between them in a computationally efficient way through another network conditioned on task pairs, and avoids learning trivial cross-task relations by retaining high-level information about the input image. We rigorously demonstrate that our proposed method effectively exploits the images with unlabelled tasks and outperforms existing semi-supervised learning approaches and related methods on three standard benchmarks.*



(a) Fully annotated dataset.

(b) Partially annotated dataset.

(c) Cross-task consistency learning.

Figure 1. **Multi-task partially supervised learning.** We look at the problem of learning multiple tasks from partially annotated data (b) where not all the task labels are available for each image, which generalizes over the standard supervised learning (a) where all task labels are available. We propose a MTL method that employs a shared feature extractor (f_ϕ) with task-specific heads (h_ψ) and exploits label correlations between each task pair by mapping them into a *joint pairwise task-space* and penalizing inconsistencies between the provided ground-truth labels and predictions (c).

1. Introduction

With the recent advances in dense prediction computer vision problems [17, 25, 38, 43, 47, 54, 57, 64, 65, 68, 72, 73], where the aim is to produce pixel-level predictions (*e.g.* semantic and instance segmentation, depth estimation), the interest of the community has started to shift towards the more ambitious goal of learning multiple of these problems jointly by multi-task learning (MTL) [7]. Compared to the standard single task learning (STL) that focuses on learning an individual model for each task, MTL aims at learning a single model for multiple tasks with a better efficiency and generalization tradeoff while sharing information and computational resources across them.

Recent MTL dense prediction methods broadly focus on designing MTL architectures [4, 5, 19, 24, 37, 39, 44, 49, 58, 60, 66, 76–78] that enable effective sharing of information across tasks and improving the MTL optimization [12, 13, 21, 23, 30, 33, 34, 37, 51, 67] to balance the influence of each task-specific loss function and to prevent interference between the tasks in training. We refer to [59] for a more comprehensive review. One common and strong assumption in these works is that each training image has to be labelled for all the tasks (Fig. 1(a)). There are two main practical limitations to this assumption. First, curating multi-task image datasets (*e.g.* KITTI [20] and CityScapes [15]) typically involves using multiple sensors to produce ground-truth labels for several tasks, and obtaining all the labels for each image requires very accurate synchronization between the sensors, which is a challenging research problem by itself [61]. Second, imagine a scenario where one would like to add a new task to an existing image dataset which is already annotated for another task and obtaining the ground-truth labels for the new task requires using a different sensor (*e.g.* depth camera) to the one which is used to capture the original data. In this case, labelling the previously recorded images for the new task will not be possible for many visual scenes (*e.g.* uncontrolled outdoor environments). Such real-world scenarios lead to obtaining partially annotated data and thus ask for algorithms that can learn from such data.

In this paper, we look at a more realistic and general case of the MTL dense prediction problem where not all the task labels are available for each image (Fig. 1(b)) and we call this setting *multi-task partially supervised learning*. In particular, we assume that each image is at least labelled for one task and each task at least has few labelled images and we would like to learn a multi-task model on them. A naive way of learning from such partial supervision is to train a multi-task model only on the available labels (*i.e.* by setting the weight of the corresponding loss function to 0 for the missing task labels). Though, in this setting, the MTL model is trained on all the images thanks to the parameter sharing across the tasks, it cannot extract the task-specific information from the images for the unlabelled tasks. To this end, one can extend existing single-task semi-supervised learning methods to MTL by penalizing the inconsistent predictions of images over multiple perturbations for the unlabelled tasks (*e.g.* [14, 29, 32, 36, 56]). While this strategy ensures consistent predictions over various perturbations, it does not guarantee consistency across the related tasks.

An orthogonal information that has recently been used in MTL is cross-task relation [40, 50, 69] which aims at producing consistent predictions across task pairs. Unfortunately existing methods are not directly applicable for learning from partial supervision, as they require either each training image to be labelled with all the task labels [50, 69] or cross-task relations that can be analytically derived [40]. In our setting, compared to [40, 50, 69], there are fewer training images available with ground-truth labels of each task pair and thus it is harder to learn the relationship. In addition, unlike [40], we focus on the general setting where one task label cannot be accurately obtained from another (*e.g.* from semantic segmentation to depth) and hence learning exact mappings between two task labels is not possible.

Motivated by these challenges, we propose a MTL approach that shares a feature extractor between tasks and also learns to relate each task pair in a learned *joint pairwise task-space* (illustrated in Fig. 1(c)), which encodes only the shared information between them and does not require the ill-posed problem of recovering labels of one task from another one. There are two challenges to this goal. First, a naive learning of the joint pairwise task-spaces can lead to trivial mappings that take all predictions to the same point such that each task produces artificially consistent encodings with each other. To this end, we regulate learning of each mapping by penalizing its output to retain high-level information about the input image. Second, the computational cost of modelling each task pair relation can get exponentially expensive with the number of tasks. To address this challenge, we use a single encoder network to learn all the pairwise-task mappings, however, dynamically estimate its weights by conditioning them on the target task pair.

The main contributions of our method are as following. We propose a new and practical setting for multi-task dense prediction problems and a novel MTL model that penalizes cross-task consistencies between pairs of tasks in joint pairwise task-spaces, each encoding the commonalities between pairs, in a computationally efficient manner. We show that our method can be incorporated to several architectures and significantly outperforms the related baselines in three standard multi-task benchmarks.

2. Related Work

Multi-task Supervised Learning. Multi-task Learning (MTL) [7, 48, 59, 74] aims at learning a single model that can infer all desired task outputs given an input. The prior works can be broadly divided into two groups. The first one [4–6, 19, 24, 37, 39, 44, 49, 58, 60, 66, 76–78] focuses on improving network architecture by better sharing information across tasks and learning task-specific representation by devising cross-task attention mechanism [44], task-specific attention modules [37], gating strategies [5, 24], etc. The second one aims to improve the imbalanced optimization problem caused by jointly optimizing different losses of various tasks as the difficulty levels, loss magnitudes, and characteristics of tasks are various. To this end, the recent work [12, 13, 21, 23, 30, 34, 37, 51, 67] enable a more balanced optimization for multi-task learning network by dynamically adjusting weights of each loss functions based on task-certainty [30], Pareto optimality [51], discarding conflicting gradient components [67], etc. However, these works focus on the supervised setting, where each sample in the dataset is annotated for all desired tasks.

Multi-task Semi-supervised Learning. Learning multi-task model on fully annotated data would require large-scale labeled data and it is costly to collect sufficient labeled data. Thus few works propose to learn multi-task learning model using semi-supervised learning strategy [14, 29, 32, 36, 36, 56, 62, 75] and they assume that the dataset consists of limited data annotated with all tasks labels and a large amount of unlabeled data. Liu *et al.* [36] extend single-task semi-supervised learning to multi-task learning by learning a classifier per task jointly under the constraint of a soft-sharing prior imposed over the parameters of the classifiers. In [14, 29, 32, 36, 56], the authors employ a regularization term on the unlabeled samples of each tasks that encourages the model to produce ‘consistent’ predictions when its inputs are perturbed.

Cross-task Relations. A rich body of work [3, 8, 27, 28, 35, 40, 41, 50, 55, 63, 69–71, 79] study the relations between tasks in MTL. Most related to ours, [50] explore the relations between segmentation and depth and propose a better fusion strategy to fuse two tasks predictions for domain adaptation. Zamir *et al*. [69] study the cross-task consistency learned from groundtruth of all tasks for robust learning, *i.e*. the predictions made for multiple tasks from the same image are not independent, and therefore, are expected to be 'consistent'. Similar to [69], Lu *et al*. [40] propose to leverage the cross-task consistency between predictions of different tasks on unlabeled data in a mediator dataset when jointly learning multiple models for distributed training. To regularize the cross-task consistency, Lu *et al*. [40] design consistency losses according to the consistency between adjacent frames in videos, relations between depth and surface normal, etc. In this paper, we also exploit the cross-task consistency in MTL, however, from partially annotated data where the mapping from one task label to another cannot be analytically derived or exactly learned. To this end, unlike [40, 69], we learn a joint task-space for each task pair rather than measuring consistency in one's task space. Finally, our method learns cross-task in a more computationally efficient way than [40, 69] by sharing parameters across different mappings and conditioning its output on the related task-pair.

3. Method

3.1. Problem setting

Let $\boldsymbol{x} \in \mathbb{R}^{3\times H\times W}$ and $\boldsymbol{y}^t \in \mathbb{R}^{O^t\times H\times W}$ denote an $H \times W$ dimensional RGB image and its dense label for task t respectively, where O^t is the number of output channels for task t. Our goal is to learn a function $\hat{y}^t$ for each task t that accurately predicts the ground-truth label $\boldsymbol{y}^t$ of previously unseen images. While such a task-specific function can be learned for each task independently, a more efficient design is to share most of the computations across the tasks via a common feature encoder, convolutional neural network $f_\phi : \mathbb{R}^{3\times H\times W} \rightarrow \mathbb{R}^{C\times H'\times W'}$ parameterized by ϕ that takes in an image and produces a C feature maps, each with $H' \times W'$ resolution, where typically $H' < H$ and $W' < W$. In this setting, f_ϕ is followed by multiple task-specific decoders $h_{\psi^t} : \mathbb{R}^{C\times H'\times W'} \rightarrow \mathbb{R}^{O^t\times H\times W}$, each with its own task-specific weights ψ^t that decodes the extracted feature to predict the label for the task t, *i.e*. $\hat{y}^t(\boldsymbol{x}) = h_{\psi^t} \circ f_\phi(\boldsymbol{x})$ (Fig. 2(a)).

Let $\mathcal{D}$ denote a set of N training images with their corresponding labels for K tasks. Assume that for each training image $\boldsymbol{x}$, we have ground-truth labels available only for some tasks where we use $\mathcal{T}$ and $\mathcal{U}$ to store the indices of labeled and unlabelled tasks respectively, where $|\mathcal{T}| + |\mathcal{U}| = K$, $\mathcal{U} = \varnothing$ indicates all labels available for $\boldsymbol{x}$ and $\mathcal{T} = \varnothing$ indicates no labels available for $\boldsymbol{x}$. In this paper, we focus on the partially annotated setting, where each image is labelled at least for one task ($|\mathcal{T}| \geq 1$) and each task at least has few labelled images.

A naive way of learning $\hat{y}^t$ for each task on the partially annotated data $\mathcal{D}$ is to jointly optimize its parameters on the labelled tasks as following:

$$\min_{\phi,\psi} \frac{1}{N}\sum_{n=1}^{N}\frac{1}{|\mathcal{T}_n|}\sum_{t\in\mathcal{T}_n} L^t(\hat{y}^t(\boldsymbol{x}_n), \boldsymbol{y}^t_n), \tag{1}$$

where n is the image index and L^t is the task-specific differentiable loss function. We denote this setting as the (vanilla) MTL. Here, thanks to the parameter sharing through the feature extractor, its task-agnostic weights are learned on all the images. However, the task-specific weights ψ^t are trained only on the labeled images.

A common strategy to exploit such information from unlabeled tasks is to formulate the problem in a semi-supervised learning (SSL) setting. Recent successful SSL techniques [2, 53] focus on learning models that can produce consistent predictions for unlabelled images when its input is perturbed in various ways.

$$\begin{aligned}\min_{\phi,\psi} \frac{1}{N}\sum_{n=1}^{N}\Big(&\frac{1}{|\mathcal{T}_n|}\sum_{t\in\mathcal{T}_n} L^t(\hat{y}^t(\boldsymbol{x}_n), \boldsymbol{y}^t_n) \\ &+ \frac{1}{|\mathcal{U}_n|}\sum_{t\in\mathcal{U}} L_u(e_r(\hat{y}^t(\boldsymbol{x}_n)), \hat{y}^t(e_r(\boldsymbol{x}_n)))\Big),\end{aligned} \tag{2}$$

where L_u is the unsupervised loss function and e_r is a geometric transformation (*i.e*. cropping) parameterized by the random variable r (*i.e*. bounding box location). In words, for the unsupervised part, we apply our model to the original input $\boldsymbol{x}$ and also its cropped version $e_r(\boldsymbol{x})$, and then we also crop the prediction corresponding to the original input $e_r(\hat{y}^t(\boldsymbol{x}_n))$ before we measure the difference between two by using L_u. Note that we are aware of more sophisticated task-specific SSL methods for semantic segmentation [42, 45], depth estimation [22, 31], however, combining them for multiple tasks, each with different network designs and learning formulations is not trivial and here we focus on one SSL strategy that uses one perturbation type (*i.e*. random cropping) and L_u (*i.e*. mean square error) can be applied to several tasks.

3.2. Cross-task consistency learning

While optimizing Eq. (2) allows learning both task-agnostic and task-specific weights on the labeled and unlabelled data, it does not leverage cross-task relations, which can be used to further supervise unlabelled tasks. Prior works [40, 69] define the cross-task relations by a mapping function $m^{s\rightarrow t}$ for each task-pair (s, t) which maps the prediction for the source task s to target task t labels. The mapping function in [40] is analytical based on the assumption that target task labels can be analytically computed from source labels. While such analytical relations is possible

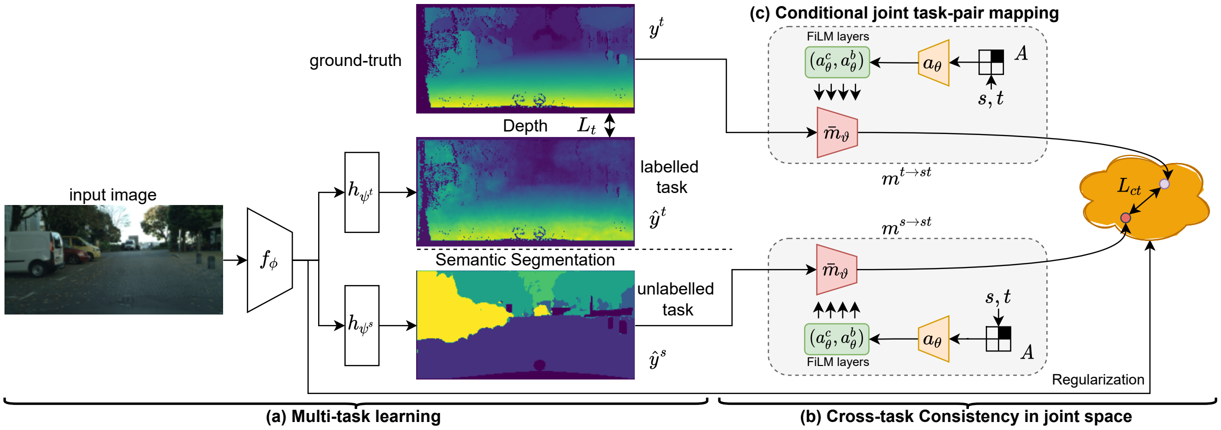


Figure 2. Illustration of our method for multi-task partially supervised learning. Given an image, our method uses a shared feature extractor f_ϕ taking in the input image and task-specific decoders (h_{ψ^s} and h_{ψ^t}) to produce predictions for all tasks (a). We compute the supervised loss L_t for labelled task. Besides, we regularize the cross-task consistency L_{ct} between the unlabelled task's prediction $\hat{\boldsymbol{y}}^s$ and the labelled task's ground-truth $\boldsymbol{y}^t$ in a joint space for the unlabelled task (b). To learn the cross-task consistency efficiently, we propose to use a shared mapping function whose output is conditioned on the task-pair (c) and regularize the learning of mapping function using the feature from f_ϕ to prevent trivial solution.

only for certain task pairs, each mapping function in [69] is parameterized by a deep network and its weights are learned by minimizing $L_{ct}(m^{s\to t}(\boldsymbol{y}^s), \boldsymbol{y}^t)$, where L_{ct} is cross-task function that measures the distance between the mapped source labels and target labels. There are two limitations to this method in our setting. First the training set has limited labelled number of images for both source and target tasks ($\boldsymbol{y}^s$ and $\boldsymbol{y}^t$). Second learning such pairwise mappings accurately is not often possible in our case, as the labels of one task can only be partially recovered from another task (*e.g*. semantic segmentation to depth estimation). Note that this ill-posed problem can be solved accurately when strong prior knowledge about the data is available.

To employ cross-task consistency to our setting, we map each task pair (s, t) to a lower-dimensional joint pairwise task-space where only the common features of both tasks are encoded (Fig. 2(b)). Formally, each pairwise task-space for (s, t) is defined by a pair of mapping functions, $m_{\vartheta^{st}_s} : \mathbb{R}^{O^s\times H\times W} \to \mathbb{R}^D$ and $m_{\vartheta^{st}_t} : \mathbb{R}^{O^t\times H\times W} \to \mathbb{R}^D$ parameterized by ϑ^{st}_s and ϑ^{st}_t respectively. The cross-task consistency can be incorporated to Eq. (1) as following:

$$\min_{\phi,\psi,\vartheta} \frac{1}{N}\sum_{n=1}^{N}\Big(\frac{1}{|\mathcal{T}_n|}\sum_{t\in\mathcal{T}_n} L^t(\hat{y}^t(\boldsymbol{x}_n), \boldsymbol{y}^t_n) + \\ \frac{1}{|\mathcal{U}_n|}\sum_{s\in\mathcal{U}_n, t\in\mathcal{T}_n} L_{ct}\big(m_{\vartheta^{st}_s}(\hat{y}^s(\boldsymbol{x}_n)), m_{\vartheta^{st}_t}(\boldsymbol{y}^t_n)\big)\Big), \tag{3}$$

where L_{ct} is cosine distance (*i.e*. $L_{ct}(\mathbf{a}, \mathbf{b}) = 1 - (\mathbf{a}\cdot\mathbf{b})/(|\mathbf{a}||\mathbf{b}|)$. In words, along with the MTL optimization, Eq. (3) minimizes the cosine distance between the embeddings of the unlabelled task prediction $\hat{\boldsymbol{y}}_s$ and the annotated task label $\boldsymbol{y}^t$ in the joint pairwise task space. Here $m_{\vartheta^{st}_s}$ and $m_{\vartheta^{st}_t}$ are not necessarily equal to allow for treating the mapping from predicted and ground-truth labels differently. Note that one can also include the semi-supervised term L_u in Eq. (3). However we empirically found that it does not bring any tangible performance gain when used with the cross-task term L_{ct}.

There are two challenges to learn non-trivial pairwise mapping functions in a computationally efficient way. First the number of pairwise mappings to learn quadratically grows with the number of tasks. Although the mapping functions are only used in training, it can still be computationally expensive to train many of them jointly. In addition, learning an accurate mapping for each task-pair can be challenging in case of limited labels. Second the mapping functions can simply learn a trivial solution such that each task is mapped to a fixed point (*e.g*. zero vector) in the joint space.

Conditional joint task-pair mapping. To address the first challenge, as shown in Fig. 2(c), we propose to use a task-agnostic mapping function $\bar{m}_\vartheta$ with one set of parameters ϑ whose output is conditioned both on the input task (s or t) and task-pair (s, t) through an auxiliary network (a_θ). Concretely, let A denote a variable that includes the input task (s or t) and target pair (s, t) for a pairwise mapping which in practice we encode with an asymmetric $K \times K$ dimensional matrix by setting the corresponding entry to 1 (*i.e*. $A[s, t] = 1$ or $A[t, s] = 1$) and the other entries to 0. Note that the diagonal entries are always zero, as we do not define any self-task relation. Let $\bar{m}_\vartheta$ be a multi-layer network and $\boldsymbol{h}_i$ denote a M channel feature map of its i-th layer for which the auxiliary network a_θ, parameterized by θ, takes in A and outputs two M-dimensional vectors $a^c_{\theta,i}$ and $a^b_{\theta,i}$. These vectors are applied to transform the feature map $\boldsymbol{h}_i$ in a similar way to [46] as following:

$$\boldsymbol{h}_i \leftarrow a^c_{\theta,i}(A) \odot \boldsymbol{h}_i + a^b_{\theta,i}(A)$$

where $\odot$ denote a Hadamard product. In words, the auxiliary network alters the output of the task-agnostic mapping function $\bar{m}_\vartheta$ based on A. For brevity, we denote the conditional mapping from s to (s,t) as $m^{s\to st}$ which is a function of $\bar{m}_\vartheta$ and a_θ and hence parameterized with ϑ and θ.

We implement each a^c_i and a^b_i as an one layer fully-connected network. Hence, given the light-weight auxiliary network, the computational load for computing the conditional mapping function, in practice, does not vary with the number of task-pairs. Finally, as the dimensionality of each task label vary – *e.g.* while O^t is 1 for depth estimation and O^t equals to number of categories in semantic segmentation –, we use task-specific input layers and pass each prediction to the corresponding one before feeding it to the joint pairwise task mapping. In the formulation, we include these layers in our mapping $\bar{m}_\vartheta$ and explain their implementation details in Sec. 4.

Regularizing mapping function. To avoid learning trivial mappings, we propose a regularization strategy (Fig. 2) that encourages the mapping to retain high-level information about the input image. To this end, we penalize the distance between the output of the mapping function and a feature vector that is extracted from the input image. In particular, we use the output of the task-agnostic feature extractor $f_\phi(\boldsymbol{x})$ in the regularization. Now we can add the regularizer to the formulation in Eq. (3):

$$\begin{aligned}\min_{\phi,\psi,\vartheta,\theta} \frac{1}{N}\sum_{n=1}^{N}\Big(&\frac{1}{|\mathcal{T}_n|}\sum_{t\in\mathcal{T}_n} L^t\big(\hat{y}^t(\boldsymbol{x}_n),\boldsymbol{y}^t_n\big)+\\ &\frac{1}{|\mathcal{U}_n|}\sum_{s\in\mathcal{U}_n,t\in\mathcal{T}_n} L_{ct}\big(m^{s\to st}(\hat{y}^s(\boldsymbol{x}_n)), m^{t\to st}(\boldsymbol{y}^t_n)\big)\\ &+R(f_\phi(\boldsymbol{x}_n), m^{s\to st}(\hat{y}_s(\boldsymbol{x}_n)))\\ &+R(f_\phi(\boldsymbol{x}_n), m^{t\to st}(\boldsymbol{y}^t_n))\Big),\end{aligned} \tag{4}$$

where $f_\phi(\boldsymbol{x})$ is the feature from feature encoder f_ϕ, R is the loss function and we use the cosine similarity loss for R in this work.

Alternative mapping strategies. Here we discuss two different mapping strategies to exploit cross-task consistency proposed in [69] and their adoption to our setting. As both require learning a mapping from one task's groundtruth label to another one and we have either no or few images with both groundtruth labels, here we approximate them by learning mappings from prediction of one task to another task's groundtruth. In the first case, one can substitute our cross-consistency loss and regularization terms with $L_{ct}(m^{s\to t}(\hat{y}^s(\boldsymbol{x})), \boldsymbol{y}^t)$ in Eq. (4), which is denoted as *Direct-Map*. In the second case, we replace our terms with $L_{ct}(m^{s\to t}(\hat{y}^s(\boldsymbol{x})), m^{s\to t}(\boldsymbol{y}^s))$ that maps both the groundtruth $\boldsymbol{y}^s$ and predicted labels $\hat{\boldsymbol{y}}^s$ and minimize their distance in task t's label space. We denote this setting as *Perceptual-Map* and compare to them in Sec. 4.

Alternative loss and regularization strategies. Alternatively, our cross-consistency loss and regularization terms can be replaced with another loss function only that does not allow for learning of trivial mappings. One such loss function is contrastive loss where one can define the predictions for two tasks on the same image as a positive pair (*i.e.* $m^{s\to st}(\hat{y}^s(\boldsymbol{x}_i))$ and $m^{t\to st}(\boldsymbol{y}^t_i)$) and on different images as a negative pair (*i.e.* $m^{s\to st}(\hat{y}^s(\boldsymbol{x}_j))$ and $m^{t\to st}(\boldsymbol{y}^t_i)$), and penalize when the distance from the positive one is bigger than the negative one. We denote this setting as *Contrastive-Loss*. Another method which also employs positive and negative pairs involves using a discriminator network. The discriminator (a convolutional neural network) takes in positive and negative pairs and predicts their binary labels, while the parameters of the MTL network and mapping functions are alternatively optimized. We denote this setting as *Discriminator-Loss* and compare to the alternative methods in Sec. 4.

4. Experiments

Datasets. We evaluate all methods on three standard dense prediction benchmarks, Cityscapes [15], NYU-V2 [52], and PASCAL [18]. Cityscapes [15] consists of street-view images, which are labeled for two tasks: 7-class semantic segmentation[1] and depth estimation. We resize the images to 128×256 to speed up the training as [37]. NYU-V2 [52] contains RGB-D indoor scene images, where we evaluate performances on 3 tasks, including 13-class semantic segmentation, depth estimation, and surface normals estimation. We use the true depth data recorded by the Microsoft Kinect and surface normals provided in [16] for depth estimation and surface normal estimation. All images are resized to 288×384 resolution as in [37]. PASCAL [18] is a commonly used benchmark for dense prediction tasks. We use the data splits from PASCAL-Context [11] which has annotations for semantic segmentation, human part segmentation and semantic edge detection. Additionally, as in [59], we also consider the tasks of surface normals prediction and saliency detection and use the annotations provided by [59].

Experimental setting. For the evaluation of multi-task models learned in different partial label regimes, we design two settings: *(i) random* setting where, we randomly select and keep labels for at least 1 and at most $K-1$ tasks where K is the number of tasks, *(ii) one* label setting, where we randomly select and keep label only for 1 task for each training image.

[1] The original version of Cityscapes provides labels 19-class semantic segmentation. We follow the evaluation protocol in [37], we use labels of 7-class semantic segmentation. Please refer to [37] for more details.

In Cityscapes and NYU-v2, we follow the training and evaluation protocol in [37] and we use the the SegNet [1] as the MTL backbone for all methods. As in [37], we use cross-entropy loss for semantic segmentation, l1-norm loss for depth estimation in Cityscapes, and cosine similarity loss for surface normal estimation in NYU-v2. We use the exactly same hyper-parameters including learning rate, optimizer and also the same evaluation metrics, mean intersection over union (mIoU), absolute error (aErr) and mean error (mErr) in the predicted angles to evaluate the semantic segmentation, depth estimation and surface normals estimation task, respectively in [37]. We use the encoder of SegNet for the joint pairwise task mapping ($\bar{m}_\vartheta$) and one convolutional layer as task specific input layer in $\bar{m}_\vartheta$. For `Direct-Map` and `Perceptual-Map`, as in [69] we use the whole SegNet as the cross-task mapping functions.

In PASCAL, we follow the training, evaluation protocol and implementation in [59] and employ the ResNet-18 [26] as the encoder shared across all tasks and Atrous Spatial Pyramid Pooling (ASPP) [10] module as task-specific heads. We use the same hyper-parameters, *e.g.* learning rate, augmentation, loss functions, loss weights in [59]. For evaluation metrics, we use the optimal dataset F-measure (odsF) [41] for edge detection, the standard mean intersection over union (mIoU) for semantic segmentation, human part segmentation and saliency estimation are evaluated, mean error (mErr) for surface normals. We modify the ResNet-18 to have task specific input layers (one convolutional layer for each task) before the residual blocks as the mapping function $\bar{m}_\vartheta$ in our method. We refer to the supplementary for more details.

4.1. Results

We compare our method to multiple baselines including the vanilla MTL Supervised Learning (SL) baseline in Eq. (1) on both all the labels and partial labels in Eq. (1), and the MTL Semi-supervised Learning (SSL) in Eq. (2), also variations of our method with Direct-Map, Perceptual-Map, Contrastive-Loss and Discriminator-Loss as described in Sec. 3. We use uniform weights for task-specific losses for all, unless stated otherwise.

Results on Cityscapes. We first compare our method to the baselines on Cityscapes in Tab. 1 for only *one* label setting as there are two tasks in total. The results of MTL model learned with SL when all task labels are available for training to serve as a strong baseline. In the partial label setting (one task label per image), the performance of the SL baseline drops substantially compared to its performance in full supervision setting. While the SSL baseline, by extracting task-specific information from unlabelled tasks, improves over SL, further improvements are obtained by exploiting cross-task consistency in various ways except Discriminator-Loss. The methods learn mappings from one task to another one (Perceptual-Map and Direct-Map) surprisingly perform better than the ones learning joint space mapping functions (Contrastive-Loss and Discriminator-Loss), possibly due to insufficient number of negative samples. Due to the same reason, we exclude the further comparisons to Contrastive-Loss and Discriminator-Loss in NYU-v2 and PASCAL and include them in the supplementary. Finally, the best results are obtained with our method that can exploit cross-task relations more efficiently through joint pairwise task mappings with the proposed regularization. Interestingly, our method also outperforms the SL baseline that has access to all the task labels, showing the potential information in the cross-task relations.

# label	Method	Seg. (IoU) ↑	Depth (aErr) ↓
full	Supervised Learning	73.36	0.0165
one	Supervised Learning	69.50	0.0186
	Semi-supervised Learning	71.67	0.0178
	Perceptual-Map	72.82	0.0169
	Direct-Map	72.33	0.0179
	Contrastive-Loss	71.79	0.0183
	Discriminator-Loss	68.94	0.0208
	Ours	**74.90**	**0.0161**

Table 1. Multi-task learning results on Cityscapes. ‘one’ indicates each image is randomly annotated with one task label.

Results on NYU-v2. We then evaluate our method along with the baselines on NYU-v2 in the *random* and *one* label settings in Tab. 2. While we observe a similar trend across different methods, overall the performances are lower in this benchmark possibly due to fewer training images than CityScapes. As expected, the performance in random-label setting is better than the one in one-label setting, as there are more labels available in the former. While the best results are obtained with SL trained on the full supervision, our method obtains the best performance among the partially supervised methods. Here SSL improves over SL trained on the partial labels and cross-task consistency is beneficial except for Direct-Map in the one label setting, possibly because the dataset is too small to learn accurate mappings between two tasks, while our method is more data-efficient and more successful to exploit the cross-task relations.

# labels	Method	Seg. (IoU) ↑	Depth (aErr) ↓	Norm. (mErr) ↓
full	Supervised learning	36.95	0.5510	29.51
random	Supervised Learning	27.05	0.6624	33.58
	Semi-supervised Learning	29.50	0.6224	33.31
	Perceptual-Map	32.20	0.6037	32.07
	Direct-Map	29.17	0.6128	33.63
	Ours	**34.26**	**0.5787**	**31.06**
one	Supervised Learning	25.75	0.6511	33.73
	Semi-supervised Learning	27.52	0.6499	33.58
	Perceptual-Map	26.94	0.6342	34.30
	Direct-Map	19.98	0.6960	37.56
	Ours	**30.36**	**0.6088**	**32.08**

Table 2. Multi-task learning results on NYU-v2. ‘random’ indicates each image is annotated with a random number of task labels and ‘one’ means each image is randomly annotated with one task.

Results on PASCAL-Context. We evaluate all methods on PASCAL-Context, in both label settings, which contains wider variety of tasks than the previous benchmarks and report the results in Tab. 3. As the required number of pairwise mappings for Direct-Map and Perceptual-Map grows quadratically (20 mappings for 5 tasks), we omit these two due to their high computational cost and compare our method only to SL and SSL baselines. We see that the SSL baseline improves the performance over SL in random-label setting, however, it performs worse than the SL in one label setting, when there are 60% less labels. Again, by exploiting task relations, our method obtains better or comparable results to SSL, while the gains achieved over SL and SSL are more significant in the low label regime (one-label). Interestingly, SSL and our method obtain comparable results in random-label setting which suggests that relations across tasks are less informative than the ones in CityScape and NYUv2.

# labels	Method	Seg. (IoU) ↑	H. Parts (IoU) ↑	Norm. (mErr) ↓	Sal. (IoU) ↑	Edge (odsF) ↑
full	Supervised Learning	63.9	58.9	15.1	65.4	69.4
random	Supervised Learning	58.4	55.3	16.0	63.9	**67.8**
	Semi-supervised Learning	**59.0**	**55.8**	**15.9**	**64.0**	66.9
	Ours	**59.0**	55.6	**15.9**	**64.0**	**67.8**
one	Supervised Learning	48.0	55.6	17.2	61.5	64.6
	Semi-supervised Learning	45.0	54.0	**16.9**	**61.7**	62.4
	Ours	**49.5**	**55.8**	17.0	**61.7**	**65.1**

Table 3. Multi-task learning results on PASCAL. 'random' indicates each image is annotated with a random number of task labels and 'one' means each image is randomly annotated with one task.

4.2. Further results

Learning from partial and imbalanced task labels. So far, we considered the partially annotated setting where the number of labels for each task is similar. We further evaluate all methods in an imbalanced partially supervised setting in Cityscapes, where we assume the ratio of labels for each task are imbalanced, *e.g.* we randomly sample 90% of images to be labeled for semantic segmentation and only 10% images having labels for depth and we denote this setting by the label ratio between segmentation and depth (Seg.:Depth = 9:1). The opposite case (Seg.:Depth = 1:9) is also considered.

#labels	Method	Seg. (IoU) ↑	Depth (aErr) ↓
full	Supervised Learning	73.36	0.0165
1:9	Supervised Learning	63.37	0.0161
	Semi-supervised Learning	64.40	0.0179
	Perceptual-Map	68.84	0.0141
	Direct-Map	67.04	0.0153
	Ours	**71.89**	**0.0131**
9:1	Supervised learning	72.77	0.0250
	Semi-supervised Learning	72.97	0.0395
	Perceptual-Map	73.36	0.0237
	Direct-Map	73.13	0.0288
	Ours	**74.23**	**0.0235**

Table 4. Multi-task learning results on Cityscapes. '#label' indicates the number ratio of labels for segmentation and depth, *e.g.* '1:9' means we have 10% of images annotated with segmentation labels and 90% of images have depth groundtruth.

We report the results in Tab. 4. The performance of supervised learning (SL) on the task with partial labels drops significantly. Though SSL improves the performance on segmentation, its performance on depth drops in both cases. In contrast to SL and SSL, our method and Perceptual-Map obtain better results on all tasks in both settings by learning cross-task consistency while our method obtains the best results by joint space mapping. This demonstrates that our model can successfully learn cross-task relations from unbalanced labels thanks to its task agnostic mapping function which can share parameters across multiple task pairs.

Cross-task consistency learning with full supervision. Our method can also be applied to fully-supervised learning setting where all task labels are available for each sample by mapping one task's prediction and another task's groundtruth to the joint space and measuring cross-task consistency in the joint space. We applied our method to NYU-v2 and compare it with the single task learning (STL) networks, vanilla MTL baseline, recent multi-task learning methods, *i.e.* MTAN [37], X-task [69], and several methods focusing on loss weighting strategies, *i.e.* Uncertainty [30], GradNorm [12], MGDA [51] and DWA [37] in Tab. 5.

Method	Seg. (IoU) ↑	Depth (aErr) ↓	Norm. (mErr) ↓
STL	37.45	0.6079	25.94
MTL	36.95	0.5510	29.51
MTAN [37]	39.39	0.5696	28.89
X-task [69]	38.91	0.5342	29.94
Uncertainty [30]	36.46	0.5376	27.58
GradNorm [12]	37.19	0.5775	28.51
MGDA [51]	38.65	0.5572	28.89
DWA [37]	36.46	0.5429	29.45
Ours	41.00	0.5148	28.58
Ours + Uncertainty	**41.09**	**0.5090**	**26.78**

Table 5. Multi-task fully-supervised learning results on NYU-v2. 'STL' indicates standard single-task learning and 'MTL' means the standard multi-task learning network.

MTL, MTAN, X-task and Ours are trained with uniform loss weights. We see that our method (Ours) performs better than the other methods with uniform loss weights, *e.g.* MTAN and X-task, where X-task regularizes cross-task consistency by learning perceptual loss with pre-trained cross-task mapping functions. This shows that cross-task consistency is informative even in the fully supervised case and our method is more effective for learning cross-task consistency. Compared to recent loss weighting strategies, our method (Ours) obtains better performance on segmentation and depth estimation than other methods while slightly worse on normal estimation compared with GradNorm and Uncertainty. This is because the loss weighting strategies enable a more balanced optimization of multi-task learning than uniformly loss weighting. Thus when we incorporate the loss weighing strategy of Uncertainty [30] to our method, *i.e.* (Ours + Uncertainty), our method obtains further improvement and outperforms both GradNorm and Uncertainty.

4.3. Ablation study

Here, we conduct an ablation study to evaluate the effect of task-pair conditional mapping function and the regularization in Eq. (4). To this end, we report results of our method without task-pair condition network (a_θ), denoted as 'Ours (w/o cond)' where we use a single mapping ($\bar{m}_\vartheta$) for all task pairs, and also our method without the regularization in Eq. (4), denoted as 'Ours (w/o reg)' in Tab. 6. First our full model outperforms both Ours (w/o cond) and Ours (w/o reg) which shows that both the components are beneficial. Ours (w/o cond) which employs the same mapping for all the task pairs still achieves better performance than the SL baseline. Surprisingly, even after removing the regularization, despite the performance drop, the pairwise mappings can still be regulated with a lower learning rate to avoid learning trivial mappings and it still outperforms the SL baseline.

# labels	Method	Seg. (IoU) ↑	Depth (aErr) ↓	Norm. (mErr) ↓
random	Supervised Learning	27.05	0.6624	33.58
	Ours (w/o cond)	34.13	0.5968	31.65
	Ours (w/o reg)	33.87	0.5887	31.24
	Ours	**34.26**	**0.5787**	**31.06**
one	Supervised Learning	25.75	0.6511	33.73
	Ours (w/o cond)	29.19	0.6181	32.62
	Ours (w/o reg)	28.36	0.6407	32.92
	Ours	**30.36**	**0.6088**	**32.08**

Table 6. Ablation study on NYU-v2. 'cond' indicates whether using conditional mapping function. 'reg' indicates whether we use regularization in Eq. (4).

4.4. Qualitative results

Here, we present some qualitative results and refer to the supplementary for more results.

Mapped outputs. Here, we visualize the intermediate feature maps of $m^{s \to st}$ and $m^{t \to st}$ for one example in NYU-v2 in Fig. 3 where s and t correspond to segmentation and surface normal estimation respectively. We observe that the functions map both task labels to a joint pairwise space where the common information is around object boundaries, which in turn enables the model to produce more accurate predictions for both tasks.

Predictions. Finally we show qualitative comparisons between our method, SL and SSL baselines on NYU-v2 in Fig. 4. We can see that our method produces more accurate predictions by leveraging cross-task consistency. We also provide additional experiments in supplementary.

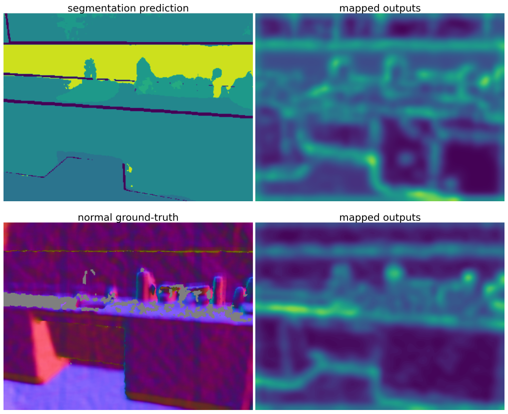


Figure 3. Intermediate feature map of the mapping function of the task-pair (segmentation to surface normal) of one example in NYU-v2. The first column shows the prediction or ground-truth and the second column present the corresponding mapped feature map (output of the mapping function's last second layer).

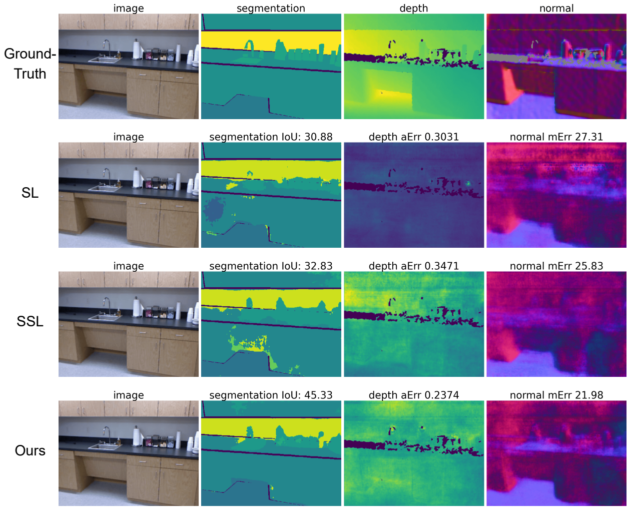


Figure 4. **Qualitative results on NYU-v2.** The fist column shows the RGB image, the second column plots the ground-truth or predictions with the IoU (↑) score of all methods for semantic segmentation, the third column presents the ground-truth or predictions with the absolute error (↓), and we show the prediction of surface normal with mean error (↓) in the last column.

5. Conclusion and Limitations

In this paper, we show that cross-task relations are crucial to learn multi-task dense prediction problems from partially annotated data in several benchmarks. We present a model agnostic method that learns relations between task pairs in joint latent spaces through mapping functions conditioned on the task pair in a computationally efficient way and also avoids learning trivial mappings with a regularization strategy. Finally, our method has limitations too. Despite the efficient learning of cross-task relations through a conditioned network, modeling cross-task relations for all task pairs may not be required. Thus it would be desirable to automatically identify which tasks are closely related and only learn such cross-task relations.

Acknowledgments. HB is supported by the EPSRC programme grant Visual AI EP/T028572/1.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *PAMI*, 39(12):2481–2495, 2017. 6, 12

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *NeurIPS*, 2019. 3

[3] Hakan Bilen and Andrea Vedaldi. Integrated perception with recurrent multi-task neural networks. In *Advances in Neural Information Processing Systems*, pages 235–243, 2016. 2

[4] Felix JS Bragman, Ryutaro Tanno, Sebastien Ourselin, Daniel C Alexander, and Jorge Cardoso. Stochastic filter groups for multi-task cnns: Learning specialist and generalist convolution kernels. In *ICCV*, pages 1385–1394, 2019. 1, 2

[5] David Bruggemann, Menelaos Kanakis, Stamatios Georgoulis, and Luc Van Gool. Automated search for resource-efficient branched multi-task networks. *arXiv preprint arXiv:2008.10292*, 2020. 1, 2

[6] David Bruggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. Exploring relational context for multi-task dense prediction. In *ICCV*, 2021. 2

[7] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997. 1, 2

[8] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI*, volume 33, pages 8001–8008, 2019. 2

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 40(4):834–848, 2017. 13

[10] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 6, 13

[11] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, pages 1971–1978, 2014. 5, 12

[12] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, pages 794–803. PMLR, 2018. 1, 2, 7, 15

[13] Zhao Chen, Jiquan Ngiam, Yanping Huang, Thang Luong, Henrik Kretzschmar, Yuning Chai, and Dragomir Anguelov. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *NeurIPS*, 2020. 1, 2

[14] Zhihao Chen, Lei Zhu, Liang Wan, Song Wang, Wei Feng, and Pheng-Ann Heng. A multi-task mean teacher for semi-supervised shadow detection. In *CVPR*, pages 5611–5620, 2020. 2

[15] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 1, 5, 12

[16] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *IEEE International Conference on Computer Vision*, pages 2650–2658, 2015. 5

[17] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014. 1

[18] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 5

[19] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *CVPR*, pages 3205–3214, 2019. 1, 2

[20] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361. IEEE, 2012. 1

[21] Ting Gong, Tyler Lee, Cory Stephenson, Venkata Renduchintala, Suchismita Padhy, Anthony Ndirango, Gokce Keskin, and Oguz H Elibol. A comparison of loss weighting strategies for multi task learning in deep neural networks. *IEEE Access*, 7:141627–141632, 2019. 1, 2

[22] Vitor Guizilini, Jie Li, Rares Ambrus, Sudeep Pillai, and Adrien Gaidon. Robust semi-supervised monocular depth estimation with reprojected distances. In *Conference on robot learning*, pages 503–512. PMLR, 2020. 3

[23] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *ECCV*, pages 270–287, 2018. 1, 2

[24] Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *ICML*, pages 3854–3863. PMLR, 2020. 1, 2

[25] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 1

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6, 12

[27] Derek Hoiem, Alexei A Efros, and Martial Hebert. Closing the loop in scene interpretation. In *CVPR*, pages 1–8. IEEE, 2008. 2

[28] Lukas Hoyer, Dengxin Dai, Qin Wang, Yuhua Chen, and Luc Van Gool. Improving semi-supervised and domain-adaptive semantic segmentation with self-supervised depth estimation. *arXiv preprint arXiv:2108.12545*, 2021. 2

[29] Abdullah-Al-Zubaer Imran, Chao Huang, Hui Tang, Wei Fan, Yuan Xiao, Dingjun Hao, Zhen Qian, and Demetri Terzopoulos. Partly supervised multitask learning. *arXiv preprint arXiv:2005.02523*, 2020. 2

[30] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491, 2018. 1, 2, 7, 13, 15

[31] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *CVPR*, pages 6647–6655, 2017. 3

[32] Siddique Latif, Rajib Rana, Sara Khalifa, Raja Jurdak, Julien Epps, and Björn W Schuller. Multi-task semi-supervised adversarial autoencoding for speech emotion recognition. *arXiv preprint arXiv:1907.06078*, 2019. 2

[33] Wei-Hong Li and Hakan Bilen. Knowledge distillation for multi-task learning. In *ECCV Workshop on Imbalance Problems in Computer Vision*, pages 163–176. Springer, 2020. 1, 13

[34] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. *NeurIPS*, 32:12060–12070, 2019. 1, 2

[35] Beyang Liu, Stephen Gould, and Daphne Koller. Single image depth estimation from predicted semantic labels. In *CVPR*, pages 1253–1260. IEEE, 2010. 2

[36] Qiuhua Liu, Xuejun Liao, and Lawrence Carin. Semi-supervised multitask learning. In *Advances in Neural Information Processing Systems*, pages 937–944, 2008. 2

[37] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *CVPR*, pages 1871–1880, 2019. 1, 2, 5, 6, 7, 12, 15

[38] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 1

[39] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *CVPR*, pages 5334–5343, 2017. 1, 2

[40] Yao Lu, Soren Pirk, Jan Dlabal, Anthony Brohan, Ankita Pasad, Zhao Chen, Vincent Casser, Anelia Angelova, and Ariel Gordon. Taskology: Utilizing task relations at scale. In *CVPR*, pages 8700–8709, 2021. 2, 3

[41] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, 2004. 2, 6, 13

[42] Robert Mendel, Luis Antonio De Souza, David Rauber, João Paulo Papa, and Christoph Palm. Semi-supervised segmentation based on error-correcting supervision. In *European Conference on Computer Vision*, pages 141–157. Springer, 2020. 3

[43] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *PAMI*, 2021. 1

[44] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, pages 3994–4003, 2016. 1, 2

[45] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *WACV*, pages 1369–1378, 2021. 3

[46] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 5

[47] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *CVPR*, pages 3227–3237, 2020. 1

[48] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 2

[49] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *AAAI*, volume 33, pages 4822–4829, 2019. 1, 2

[50] Suman Saha, Anton Obukhov, Danda Pani Paudel, Menelaos Kanakis, Yuhua Chen, Stamatios Georgoulis, and Luc Van Gool. Learning to relate depth and semantics for unsupervised domain adaptation. In *CVPR*, pages 8197–8207, 2021. 2

[51] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *NeurIPS*, 2018. 1, 2, 7, 15

[52] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012. 5, 12

[53] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *NeurIPS*, 2020. 3

[54] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikainen, and Li Liu. Pixel difference networks for efficient edge detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5117–5127, 2021. 1

[55] Boyang Sun, Jiaxu Xing, Hermann Blum, Roland Siegwart, and Cesar Cadena. See yourself in others: Attending multiple tasks for own failure detection. *arXiv preprint arXiv:2110.02549*, 2021. 2

[56] Demetri Terzopoulos et al. Semi-supervised multi-task learning with chest x-ray images. In *International Workshop on Machine Learning in Medical Imaging*, pages 151–159. Springer, 2019. 2

[57] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, pages 282–298. Springer, 2020. 1

[58] Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: deciding what layers to share. In *BMVC*, 2020. 1, 2

[59] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *PAMI*, 2021. 1, 2, 5, 6, 12, 13, 14

[60] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *ECCV*, pages 527–543. Springer, 2020. 1, 2

[61] Raphael Voges and Bernardo Wagner. Timestamp offset calibration for an imu-camera system under interval uncertainty. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 377–384. IEEE, 2018. 1

[62] Fei Wang, Xin Wang, and Tao Li. Semi-supervised multi-task learning with task regularizations. In *ICDM*, pages 562–568. IEEE, 2009. 2

[63] Qin Wang, Dengxin Dai, Lukas Hoyer, Luc Van Gool, and Olga Fink. Domain adaptive semantic segmentation with self-supervised depth estimation. In *ICCV*, pages 8515–8525, 2021. 2

[64] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *CVPR*, pages 8741–8750, 2021. 1

[65] Fangting Xia, Peng Wang, Xianjie Chen, and Alan L Yuille. Joint multi-person pose estimation and semantic part segmentation. In *CVPR*, pages 6769–6778, 2017. 1

[66] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*, pages 675–684, 2018. 1, 2

[67] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *NeurIPS*, 2020. 1, 2

[68] Zhiding Yu, Chen Feng, Ming-Yu Liu, and Srikumar Ramalingam. Casenet: Deep category-aware semantic edge detection. In *CVPR*, pages 5964–5973, 2017. 1

[69] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust learning through cross-task consistency. In *CVPR*, pages 11197–11206, 2020. 2, 3, 4, 5, 6, 7, 12, 15

[70] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, pages 3712–3722, 2018. 2

[71] Amir R Zamir, Tilman Wekel, Pulkit Agrawal, Colin Wei, Jitendra Malik, and Silvio Savarese. Generic 3d representation via pose estimation and matching. In *ECCV*, pages 535–553. Springer, 2016. 2

[72] Jing Zhang, Deng-Ping Fan, Yuchao Dai, Saeed Anwar, Fatemeh Sadat Saleh, Tong Zhang, and Nick Barnes. Uc-net: Uncertainty inspired rgb-d saliency detection via conditional variational autoencoders. In *CVPR*, pages 8582–8591, 2020. 1

[73] Miao Zhang, Weisong Ren, Yongri Piao, Zhengkun Rong, and Huchuan Lu. Select, supplement and focus for rgb-d saliency detection. In *CVPR*, pages 3472–3481, 2020. 1

[74] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017. 2

[75] Yu Zhang and Dit-Yan Yeung. Semi-supervised multi-task regression. In *ECML PKDD*, pages 617–631. Springer, 2009. 2

[76] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. Joint task-recursive learning for semantic segmentation and depth estimation. In *ECCV*, pages 235–251, 2018. 1, 2

[77] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *CVPR*, pages 4106–4115, 2019. 1, 2

[78] Ling Zhou, Zhen Cui, Chunyan Xu, Zhenyu Zhang, Chaoqun Wang, Tong Zhang, and Jian Yang. Pattern-structure diffusion for multi-task learning. In *CVPR*, pages 4514–4523, 2020. 1, 2

[79] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, pages 1851–1858, 2017. 2

A. Implementation Details

Tab. 7 and Tab. 8 provide an overview of the experimental settings, in particular report the number of train and test samples for each benchmark and number of labels used in different partially annotated settings respectively. Next we explain the implementation details for each dataset.

Cityscapes. The Cityscapes dataset [15] contains 3475 labelled images. As in [37], we use 2975 images for training and 500 images for testing. In multi-task partially supervised learning setting, we consider the one-label setting in Cityscapes, as there are only two tasks in total, *i.e.* we randomly select and keep label only for 1 task for each training image, resulting in 1487 training images annotated for segmentation and 1488 training images labelled for depth estimation, as shown in Tab. 8.

We follow the training and evaluation protocol in [37] and we use SegNet [1] as the MTL backbone for all methods, use cross-entropy loss for semantic segmentation, l1-norm loss for depth estimation. We use the exactly same hyperparameters including learning rate, optimizer as in [37]. More specifically, we use Adam optimizer with a learning rate of 0.0001 and train all models for 200 epochs with a batch size of 8 and halve the learning rate at the 100-th epoch. We also employ the same evaluation metrics, mean intersection over union (mIoU) and absolute error (aErr) to evaluate the semantic segmentation and depth estimation task, respectively as in [37].

For our model, we use the encoder architecture of SegNet for instantiating the joint pairwise task mapping ($\bar{m}_\vartheta$) and include one convolutional layer as task specific input layer in $\bar{m}_\vartheta$. For `Direct-Map` and `Perceptual-Map`, as in [69] we use the whole SegNet as the cross-task mapping functions. We use the same data augmentations from the updated implementation in [37][2], *i.e.* random crops and rand horizontal flips.

Dataset	# Train	# Test	Segmentation	Depth	Human Parts	Normals	Saliency	Edges
Cityscapes [15]	2975	500	✓	✓	- -	- -	- -	- -
NYU-v2 [52]	795	654	✓	✓	- -	✓	- -	- -
PASCAL [11]	4998	5105	✓	- -	✓	✓	✓	✓

Table 7. Details of multi-task benchmarks.

NYU-v2. The dataset [52] contains 795 training images and 654 test images. To evaluate the multi-task partially supervised learning, we consider one-label and random-label settings. For one-label setting, we randomly select and keep label for only 1 task for each training image, resulting in 265 images with annotation for segmentation, 265 images labelled for depth estimation and 265 images for surface normal. For random-label setting, we randomly select and keep labels for at least 1 and at most 2 tasks (1.49 labels per image), *i.e.* 392 images for semantic segmentation, 408 images for depth estimation, 385 images for surface normal, as shown in Tab. 8.

Dataset	# label	# labelled images Segmentation	Depth	Human Parts	Normals	Saliency	Edges
Cityscapes [15]	one	1487	1488	- -	- -	- -	- -
NYU-v2 [52]	random	392	408	- -	385	- -	- -
	one	265	265	- -	265	- -	- -
PASCAL [11]	random	2450	- -	2553	2480	2445	2557
	one	1000	- -	999	1000	1000	999

Table 8. Details about multi-task partially supervised learning settings in three benchmarks used in this work. 'random' means the random-label setting where each training image has a random number of task labels and 'one' indicates the one-label setting where each training image is annotated with one task label. '# labelled images' shows the number of images containing labels for each task, *e.g.* segmentation.

We follow the training and evaluation protocol in [37] and we use the the SegNet [1] as the MTL backbone for all methods. As in [37], we use cross-entropy loss for semantic segmentation, l1-norm loss for depth estimation and cosine similarity loss for surface normal estimation, use the same optimizer and hyper-parameters, *i.e.* Adam optimizer with a learning rate of 0.0001. We train the all model for 200 epochs with a batch size of 2 and halve the learning rate at the 100-th epoch and employ the same evaluation metrics, mean intersection over union (mIoU), absolute error (aErr) and mean error (mErr) in the predicted angles to evaluate the semantic segmentation, depth estimation and surface normals estimation task, respectively as in [37].

We use the encoder of SegNet architecture for the joint pairwise task mapping ($\bar{m}_\vartheta$) and one convolutional layer as task specific input layer in $\bar{m}_\vartheta$. For `Direct-Map` and `Perceptual-Map`, as in [69] we use the whole SegNet as the cross-task mapping functions. To regularize training, we use the exact same data augmentations from the updated implementation from [37], *e.g.* random crops and rand horizontal flips augmentations.

PASCAL-context. The dataset [11] contains 4998 training images and 5105 testing images for five tasks, *i.e.* semantic segmentation, human parts segmentation, surface normal, saliency detection and edge detection. We consider two partially supervised learning settings, random-label and one-label setting. For one-label setting, we have 1 label per image, *i.e.* 1000, 999, 1000, 1000, 999 labelled images for semantic segmentation, human parts, surface normal, saliency and edge detection, respectively. In random-label setting, we randomly sample and keep labels for at least 1 and at most 4 tasks (2.50 labels per image), resulting in 2450, 2553, 2480, 2445, 2557 labelled images for semantic segmentation, human parts, surface normal, saliency and edge detection, respectively, as shown in Tab. 8.

We follow exactly the same training, evaluation protocol and implementation in [59] and employ the ResNet-18 [26] as the encoder shared across all tasks and Atrous Spatial

[2] https://github.com/lorenmt/mtan

Pyramid Pooling (ASPP) [10] module as task-specific heads. We use the same hyper-parameters, *e.g.* learning rate, augmentation, loss functions, loss weights in [59]. More specifically, we use Adam as the optimizer with a learning rate of 0.0001 and a weight decay of 0.0001. As in [59] all experiments are performed using pre-trained ImageNet weights. We train all multi-task learning methods for 100 epochs with a batch size of 6 and we anneal the learning rate using the 'poly' learning rate scheduler as in [9, 59]. We follow [59] and use fixed loss weights for training all multi-task learning methods, *i.e.* the loss weight is 1, 2, 10, 5, 50 for semantic segmentation, human parts segmentation, surface normal estimation, saliency detection and edge detection, respectively. Please refer to [59] for more details. For evaluation metrics, we use the optimal dataset F-measure (odsF) [41] for edge detection, the standard mean intersection over union (mIoU) for semantic segmentation, human part segmentation and saliency estimation are evaluated, mean error (mErr) for surface normals. We modify the ResNet-18 to have task specific input layers (one convolutional layer for each task) before the residual blocks as the mapping function $\bar{m}_\vartheta$ in our method.

Multi-task performance. Following prior work [59], we also report the multi-task performance △MTL of the multi-task learning model as the average per-task drop in performance w.r.t. the single-task baseline:

$$\triangle\text{MTL} = \frac{1}{K}\sum_{t=1}^{K}(-1)^{\ell_i}(P_t^{mtl} - P_t^{stl})/P_t^{stl}, \tag{5}$$

where $\ell_i = 1$ if a lower value of P_t means better performance for metric of task t, and 0 otherwise.

B. More results

Here, we report more results from single-task learning (STL) model, Contrastive-Loss and Discriminator-Loss and also qualitative results.

B.1. Quantitative results

Results on Cityscapes. Here, we report the results on Cityscapes for only *one* label setting as there are two tasks in total in Tab. 9. We also report results of single-task learning models which are used to compute the multi-task performance (△MTL) to better analyze the results as in [59]. The performance of MTL methods are worse than single-task learning models for some tasks as the MTL models have less capacity and there is a problem of imbalanced optimization etc as discussed in [30, 33, 59].

The results of MTL model learned with SL when all task labels are available for training to serve as a strong baseline for multi-task learning methods. In the partial label setting (one task label per image), the performance of the SL baseline drops substantially compared to its performance in full supervision setting. While the SSL baseline, by extracting task-specific information from unlabelled tasks, improves over SL, further improvements are obtained by exploiting cross-task consistency in various ways except Discriminator-Loss. The methods learn mappings from one task to another one (Perceptual-Map and Direct-Map) surprisingly perform better than the ones learning joint space mapping functions (Contrastive-Loss and Discriminator-Loss), possibly due to insufficient number of negative samples. Finally, the best results (*e.g.* the best multi-task performance △MTL) are obtained with our method that can exploit cross-task relations more efficiently through joint pairwise task mappings with the proposed regularization. Interestingly, our method also outperforms the SL baseline that has access to all the task labels, showing the potential information in the cross-task relations.

# label	Type	Method	Seg. (IoU) ↑	Depth (aErr) ↓	△MTL ↑
full	STL	Supervised Learning	74.19	0.0124	+0.00
	MTL	Supervised Learning	73.36	0.0165	-17.00
one	STL	Supervised Learning	70.26	0.0141	+0.00
	MTL	Supervised Learning	69.50	0.0186	-16.55
		Semi-supervised Learning	71.67	0.0178	-12.22
		Perceptual-Map	72.82	0.0169	-8.37
		Direct-Map	72.33	0.0179	-11.94
		Contrastive-Loss	71.79	0.0183	-13.77
		Discriminator-Loss	68.94	0.0208	-24.95
		Ours	**74.90**	**0.0161**	**-3.81**

Table 9. Multi-task learning results on Cityscapes. 'one' indicates each image is randomly annotated with one task label. 'STL' means single task learning and 'MTL' indicates multi-task learning.

Results on Cityscapes with larger images. We also provide results for 256 × 512 setting in Tab. 10. Performance of all methods improve significantly compared to their ones using small images (in Tab. 9) and our method achieves significant improvement over the baselines.

# label	Type	Method	Seg. (IoU) ↑	Depth (aErr) ↓	△MTL ↑
one	STL	Supervised Learning	77.97	0.0126	+0.00
	MTL	Supervised Learning	77.71	0.0165	-15.95
		Semi-supervised Learning	79.24	0.0161	-13.38
		Ours	**82.41**	**0.0143**	**-4.08**

Table 10. Multi-task learning results on Cityscapes using 256 × 512 images. 'one' indicates each image is randomly annotated with one task label. 'STL' means single task learning and 'MTL' indicates multi-task learning.

Results on NYU-v2 Here, we evaluate our method and related methods in the *random* and *one* label settings on NYU-v2 and we report the results in Tab. 11. We also report results of single-task learning models which are used to compute the multi-task performance (△MTL) to better analyze the results as in [59].

While we observe a similar trend across different methods, overall the performances are lower in this benchmark possibly due to fewer training images than CityScapes. As expected, the performance in random-label setting is better than the one in one-label setting, as there are more labels available in the former. While the best results are obtained with SL trained on the full supervision, our method obtains the best performance (*e.g*. best results on all tasks and the best multi-task performance) among the partially supervised methods. Here SSL improves over SL trained on the partial labels and cross-task consistency is beneficial except for Direct-Map in the one label setting and Discriminator-Loss, possibly because the dataset is too small to learn accurate mappings between two tasks, while our method is more data-efficient and more successful to exploit the cross-task relations. In random-label setting, where images might have labels for more than one task, we also report our method also leveraging the labelled corss-task relations (‘Ours+’) in Tab. 11 and it can indeed further boost the average performance.

# labels	Type	Method	Seg. (IoU) ↑	Depth (aErr) ↓	Norm. (mErr) ↓	△MTL ↑
full	STL	Supervised learning	37.45	0.6079	25.94	+0.00
	MTL	Supervised learning	36.95	0.5510	29.51	-1.92
random	STL	Supervised Learning	28.72	0.7540	28.95	+0.00
	MTL	Supervised Learning	27.05	0.6624	33.58	-3.23
		Semi-supervised Learning	29.50	0.6224	33.31	+1.70
		Perceptual-Map	32.20	0.6037	32.07	+7.10
		Direct-Map	29.17	0.6128	33.63	+1.38
		Contrastive-Loss	30.75	0.6143	32.05	+4.96
		Discriminator-Loss	26.76	0.6354	33.13	-1.84
		Ours	34.26	0.5787	**31.06**	+11.81
		Ours+	**34.91**	**0.5738**	31.20	**+12.57**
one	STL	Supervised Learning	24.71	0.7666	30.14	+0.00
	MTL	Supervised Learning	25.75	0.6511	33.73	+1.14
		Semi-supervised Learning	27.52	0.6499	33.58	+3.16
		Perceptual-Map	26.94	0.6342	34.30	+2.31
		Direct-Map	19.98	0.6960	37.56	-12.86
		Contrastive-Loss	26.65	0.6387	34.69	+1.31
		Discriminator-Loss	25.68	0.6566	34.02	+0.04
		Ours	**30.36**	**0.6088**	**32.08**	**+10.24**

Table 11. Multi-task learning results on NYU-v2. ‘random’ indicates each image is annotated with a random number of task labels and ‘one’ means each image is randomly annotated with one task. ‘STL’ means single task learning and ‘MTL’ indicates multi-task learning.

Results on PASCAL. We evaluate all methods on PASCAL-Context, in both label settings, which contains wider variety of tasks than the previous benchmarks and report the results in Tab. 12. As in Cityscapes and NYU-v2, we also report results of single-task learning models which are used to compute the multi-task performance (△MTL) to better analyze the results as in [59].

As the required number of pairwise mappings for Direct-Map and Perceptual-Map grows quadratically (20 mappings for 5 tasks), we omit these two due to their high computational cost and compare our method only to SL, SSL, Contrastive-Loss and Discriminator-Loss baselines. We see that the SSL baseline improves the performance over SL in random-label setting, however, it performs worse than the SL in one label setting, when there are 60% less labels. By leveraging cross-task consistency, Contrastive-Loss and Discriminator-Loss obtains better performance than the SL baseline in one label setting while they get similar multi-task performance to the SL baseline in random label setting. Again, by exploiting task relations, our method obtains better or comparable results to second best method, *i.e*. SSL, while the gains achieved over SL and SSL are more significant in the low label regime (one-label). Interestingly, SSL and our method obtain comparable results in random-label setting which suggests that relations across tasks are less informative than the ones in CityScape and NYUv2.

# labels	Type	Method	Seg. (IoU) ↑	H. Parts (IoU) ↑	Norm. (mErr) ↓	Sal. (IoU) ↑	Edge (odsF) ↑	△MTL ↑
full	STL	Supervised Learning	66.4	58.9	13.9	66.7	68.3	+0.00
	MTL	Supervised Learning	63.9	58.9	15.1	65.4	69.4	-2.75
random	STL	Supervised Learning	60.9	55.3	14.7	64.8	66.8	+0.00
	MTL	Supervised Learning	58.4	55.3	16.0	63.9	**67.8**	-2.67
		Semi-supervised Learning	**59.0**	**55.8**	**15.9**	**64.0**	66.9	-2.44
		Contrastive-Loss	**59.0**	55.3	16.0	63.8	**67.8**	-2.44
		Discriminator-Loss	57.9	55.2	16.2	63.4	67.4	-3.35
		Ours	**59.0**	55.6	**15.9**	**64.0**	**67.8**	**-2.15**
one	STL	Supervised Learning	47.7	56.2	16.0	61.9	64.0	+0.00
	MTL	Supervised Learning	48.0	55.6	17.2	61.5	64.6	-1.34
		Semi-supervised Learning	45.0	54.0	**16.9**	**61.7**	62.4	-3.02
		Contrastive-Loss	48.5	55.4	17.1	61.3	64.6	-1.25
		Discriminator-Loss	48.2	**56.0**	17.1	61.7	64.7	-1.04
		Ours	**49.5**	55.8	17.0	**61.7**	**65.1**	**-0.40**

Table 12. Multi-task learning results on PASCAL. ‘random’ indicates each image is annotated with a random number of task labels and ‘one’ means each image is randomly annotated with one task. ‘STL’ means single task learning and ‘MTL’ indicates multi-task learning.

Learning from partial and imbalanced task labels. We also evaluate our method and baselines in an imbalanced partially supervised setting in Cityscapes, where we assume the ratio of labels for each task are imbalanced, *e.g*. we randomly sample 90% of images to be labeled for semantic segmentation and only 10% images having labels for depth and we denote this setting by the label ratio between segmentation and depth (Seg.:Depth = 9:1). The opposite case (Seg.:Depth = 1:9) is also considered. We report the results in Tab. 13, where we also report results of single-task learning models which are used to compute the multi-task performance (△MTL) to better analyze the results as in [59].

The performance of supervised learning (SL) on the task with partial labels drops significantly. Though SSL improves the performance on segmentation, its performance on depth drops in both cases. Different from SSL, Direct-Map, Contrastive-Loss and Discriminator-Loss improves the performance on both tasks in 1:9 setting while their performance on depth drop in the 9:1 case. In contrast to SL and the baselines, our method and Perceptual-Map obtain better results on all tasks in both settings by learning cross-task consistency while our method obtains the best performance (*i.e*. best results in all tasks and best multi-task performance, △MTL) by joint space mapping. This demonstrates that our model can successfully learn cross-task relations from unbalanced labels thanks to its task agnostic mapping function which can share parameters across multiple task pairs.

Cross-task consistency learning in conventional semi-supervised learning. We evaluate our method and SSL baseline on conventional SSL setting where $\frac{1}{3}$ of training

#labels	Type	Method	Seg. (IoU) ↑	Depth (aErr) ↓	△MTL ↑
full	STL	Supervised learning	74.19	0.0124	+0.00
	MTL	Supervised Learning	73.36	0.0165	-17.00
1:9	STL	Supervised learning	62.23	0.0126	+0.00
	MTL	Supervised Learning	63.37	0.0161	-13.07
		Semi-supervised Learning	64.40	0.0179	-19.36
		Perceptual-Map	68.84	0.0141	-0.68
		Direct-Map	67.04	0.0153	-6.90
		Contrastive-Loss	67.12	0.0151	-5.95
		Discriminator-Loss	68.92	0.0144	-1.80
		Ours	**71.89**	**0.0131**	**+5.63**
9:1	STL	Supervised learning	72.62	0.0191	+0.00
	MTL	Supervised learning	72.77	0.0250	-15.25
		Semi-supervised Learning	72.97	0.0395	-53.11
		Perceptual-Map	73.36	0.0237	-11.34
		Direct-Map	73.13	0.0288	-19.38
		Contrastive-Loss	73.75	0.0243	-12.86
		Discriminator-Loss	72.97	0.0248	-14.65
		Ours	**74.23**	**0.0235**	**-10.23**

Table 13. Multi-task learning results on Cityscapes. '#label' indicates the number ratio of labels for segmentation and depth, *e.g.* '1:9' means we have 10% of images annotated with segmentation labels and 90% of images have depth groundtruth. 'STL' means single task learning and 'MTL' indicates multi-task learning.

data in NYU-v2 are labeled for all tasks and $\frac{2}{3}$ are unlabeled, and report the results in Tab. 14. In this setting, our method obtains better performance than SL and SSL. We will include a more detailed analysis in the final paper.

Type	Method	Seg. (IoU) ↑	Depth (aErr) ↓	Norm. (mErr) ↓	△MTL ↑
MTL	Supervised Learning	24.78	0.6681	33.90	+1.48
	Semi-supervised Learning	26.09	0.6510	33.60	+4.37
	Ours	**28.43**	**0.6366**	**33.01**	**+8.83**

Table 14. Multi-task learning results on NYU-v2 in SSL setting where $\frac{1}{3}$ of training data in NYU-v2 are labeled for all tasks and $\frac{2}{3}$ are unlabeled. 'MTL' indicates multi-task learning.

Cross-task consistency learning with full supervision. Our method can also be applied to fully-supervised learning setting where all task labels are available for each sample by mapping one task's prediction and another task's groundtruth to the joint space and measuring cross-task consistency in the joint space. We applied our method to NYU-v2 and compare it with the single task learning (STL) networks, vanilla MTL baseline, recent multi-task learning methods, *i.e.* MTAN [37], X-task [69], and several methods focusing on loss weighting strategies, *i.e.* Uncertainty [30], GradNorm [12], MGDA [51] and DWA [37] in Tab. 15. Here, we also report the multi-task performance (△MTL) of all MTL methods.

MTL, MTAN, X-task and Ours are trained with uniform loss weights. We see that our method (Ours) performs better than the other methods with uniform loss weights, *e.g.* MTAN and X-task, where X-task regularizes cross-task consistency by learning perceptual loss with pre-trained cross-task mapping functions. This shows that cross-task consistency is informative even in the fully supervised case and our method is more effective for learning cross-task consistency. Compared to recent loss weighting strategies, our method (Ours) obtains better multi-task performance (△MTL) and better performance on segmentation and depth estimation than other methods while slightly worse on normal estimation compared with GradNorm and Uncertainty. This is because the loss weighting strategies enable a more balanced optimization of multi-task learning model than uniformly loss weighting. Thus when we incorporate the loss weighing strategy of Uncertainty [30] to our method, *i.e.* (Ours + Uncertainty), our method obtains further improvement and outperforms both GradNorm and Uncertainty, *e.g.* 'Ours + Uncertainty' obtains the best multi-task performance (+7.57).

Method	Seg. (IoU) ↑	Depth (aErr) ↓	Norm. (mErr) ↓	△MTL
STL	37.45	0.6079	25.94	+0.00
MTL	36.95	0.5510	29.51	-1.92
MTAN [37]	39.39	0.5696	28.89	+0.03
X-task [69]	38.91	0.5342	29.94	+0.89
Uncertainty [30]	36.46	0.5376	27.58	+0.86
GradNorm [12]	37.19	0.5775	28.51	-1.86
MGDA [51]	38.65	0.5572	28.89	+0.06
DWA [37]	36.46	0.5429	29.45	-1.82
Ours	41.00	0.5148	28.58	+4.88
Ours + Uncertainty	**41.09**	**0.5090**	**26.78**	**+7.57**

Table 15. Multi-task fully-supervised learning results on NYU-v2. 'STL' indicates standard single-task learning and 'MTL' means the standard multi-task learning network.

B.2. Qualitative results

Here, we present some qualitative results.

Mapped outputs. Here, we visualize the intermediate feature maps of $m^{s\rightarrow st}$ and $m^{t\rightarrow st}$ for one example in Cityscapes in Fig. 5 where s and t correspond to segmentation and depth estimation respectively and one example in NYU-v2 in Fig. 6 where s and t correspond to segmentation and surface normal estimation respectively. We observe that the functions map both task labels to a joint pairwise space where the common information is around object boundaries, which in turn enables the model to produce more accurate predictions for both tasks.

Predictions. Finally we show qualitative comparisons between our method, SL and SSL baselines, Perceptual-Map (PM), Direct-Map (DM), Contrastive-Loss (CL) and Discriminator-Loss (DL) on Cityscapes in Fig. 7 and on NYU-v2 in Fig. 8. We can see that our method produces more accurate predictions by leveraging cross-task consistency. Specifically, in Fig. 7, compared with methods that do not leverage cross-task consistency, the prediction of segmentation and depth are improved by our method (top left region) and our results are more accurate than related baselines (PM, DM, CL and DL). In Fig. 8, we can see that SSL produces more accurate predictions on segmentation and surface normal than SL. And PM obtains more accurate

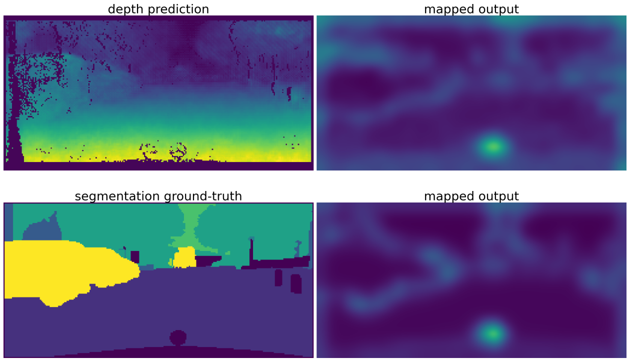


Figure 5. Intermediate feature map of the mapping function of the task-pair (segmentation to depth) of one example in Cityscapes. The first column shows the prediction or ground-truth and the second column present the corresponding mapped feature map (output of the mapping function's last second layer).

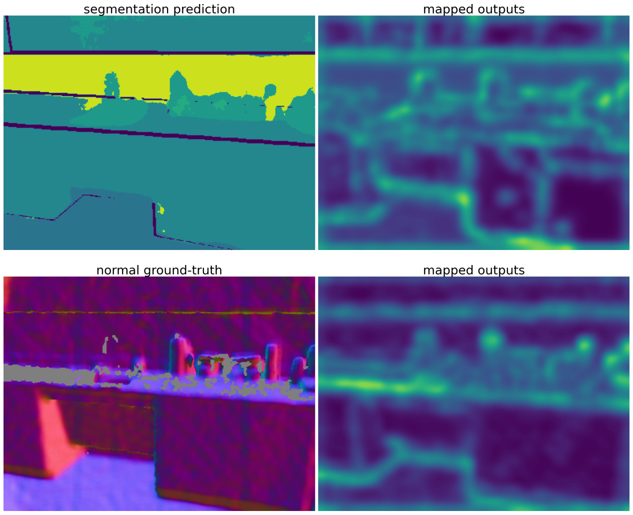


Figure 6. Intermediate feature map of the mapping function of the task-pair (segmentation to surface normal) of one example in NYU-v2. The first column shows the prediction or ground-truth and the second column present the corresponding mapped feature map (output of the mapping function's last second layer).

results on depth and surface normal than SL. While they do not achieve consistent improvement on all three tasks, our method can improve the results consistently on three tasks which shows that our method is more effective on learning cross-task consistency for MTL from partially annotated data.

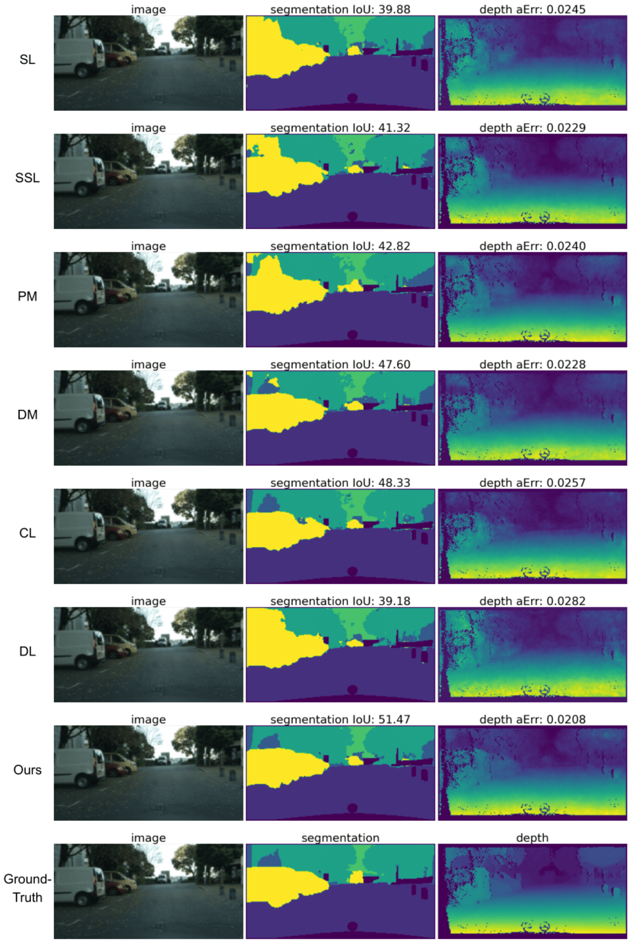


Figure 7. **Qualitative results on Cityscapes.** The fist column shows the RGB image, the second column plots the ground-truth or predictions with the IoU (↑) score of all methods for semantic segmentation and we show the ground-truth or predictions with the absolute error (↓) in the last column.

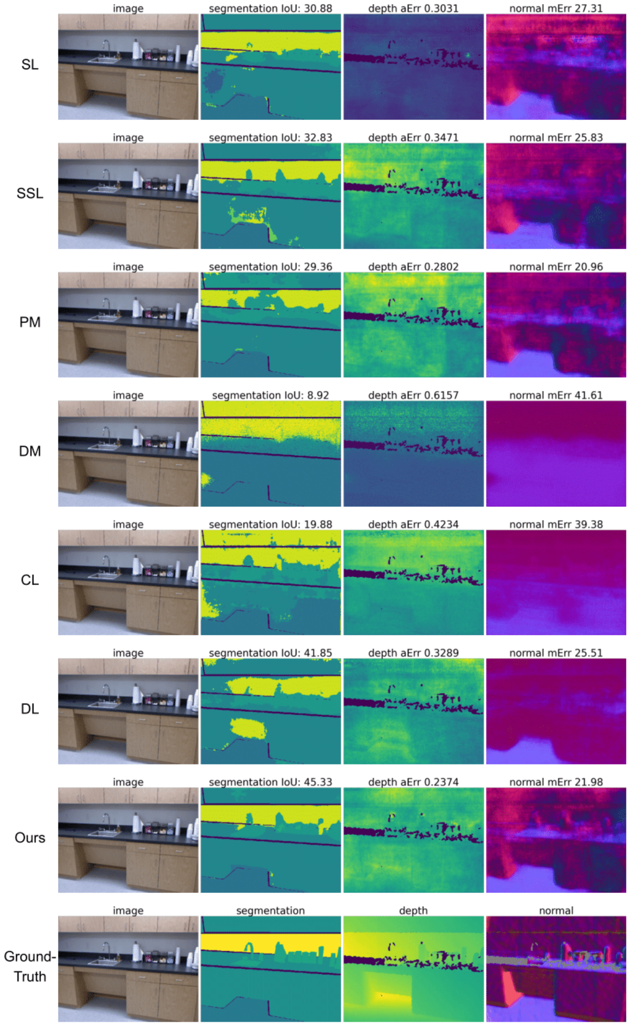


Figure 8. **Qualitative results on NYU-v2.** The fist column shows the RGB image, the second column plots the ground-truth or predictions with the IoU (↑) score of all methods for semantic segmentation, the third column presents the ground-truth or predictions with the absolute error (↓), and we show the prediction of surface normal with mean error (↓) in the last column.