

# Cross-task Attention Mechanism for Dense Multi-task Learning

Ivan Lopes  
Inria

ivan.lopes@inria.fr

Tuan-Hung Vu  
Valeo.ai, Inria

tuan-hung.vu@valeo.com

Raoul de Charette  
Inria

raoul.de-charette@inria.fr

## Abstract

*Multi-task learning has recently become a promising solution for comprehensive understanding of complex scenes. With an appropriate design, multi-task models can not only be memory-efficient but also favour the exchange of complementary signals across tasks. In this work, we jointly address 2D semantic segmentation, and two geometry-related tasks, namely dense depth, surface normal estimation as well as edge estimation showing their benefit on several datasets. We propose a novel multi-task learning architecture that exploits pair-wise cross-task exchange through correlation-guided attention and self-attention to enhance the average representation learning for all tasks. We conduct extensive experiments on three multi-task setups, showing the benefit of our proposal in comparison to competitive baselines in both synthetic and real benchmarks. We also extend our method to the novel multi-task unsupervised domain adaptation setting. Our code is available at <https://github.com/cv-rits/DenseMTL>*

## 1. Introduction

Recent advances in deep neural network architectures [12, 14] and efficient optimization techniques [1, 16] constantly set higher accuracy in scene understanding tasks, demonstrating great potentials for autonomous applications. Nevertheless, the majority of the literature focuses on pushing performance of single-tasks, being either semantic tasks like segmentation [4, 13] and detection [32], or geometrical tasks like depth/normal estimation [19, 10]. Very few have paid attention to a more comprehensive objective of joint semantics and geometry understanding, while in practice, this is desirable in critical applications such as robotics and autonomous driving. In those systems, we would expect to have cooperative synergies among all tasks, *i.e.* tasks should be processed together in one unified system rather than separately. Arguably, promoting such synergies could bring mutual benefit to all tasks involved. For example, disruptive changes in depth maps may signal semantic boundaries in segmentation maps; while pixels of some semantic classes,

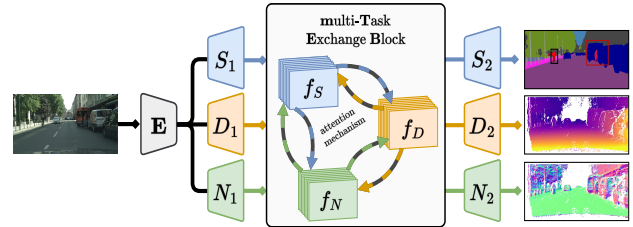


Figure 1: **Overview of our MTL framework.** The three tasks semantic Segmentation, Depth regression and Normal estimation share an encoder  $E$ . The task-specific decoders  $S$ ,  $D$  and  $N$  exchange information in the “multi-Task Exchange Block” (mTEB) via an attention-based mechanism, resulting in refined features to produce final predictions.

like “road”, may share similar surface normals.

To this end, multi-task learning (MTL) [3, 17, 26] has become a promising solution as it seeks to learn a unified model that excels in average on *all* tasks. A common MTL design is to have a large amount of parameters shared among tasks while keeping certain private parameters for individual ones; information is exchanged via shared parameters, making possible the synergy flow. Some recent works [2, 27, 33] focus on new multi-modal modules that boost tasks interactions. One important advantage of such MTL models is memory-efficiency thanks to the shared parts. In this work, we adopt the same principle and design our model with shared encoder and dedicated task decoders.

Recent MTL approaches come with different solutions to improve the information exchanged or distilled across tasks, often known as *multi-modal distillation*. PAD-Net [33] and MTI-Net [27] have proven the effectiveness of self-attention [29] for MTL, *i.e.* a mechanism to self-discover most attentive signals of each task-specific feature for the other tasks. Differently, ATRC [2] advocates to better look at pair-wise task similarity to guide multi-modal distillation. We partially follow the same direction as in ATRC with our *correlation-guided attention*, in which we use correlations between task-specific features to guide the construction of exchanged messages. Further, we propose an unified mechanism that combines two different attentions, *correlation-*

*guided attention* and *self-attention*, via a learnable channel-wise weighting scheme.

In summary, we propose a novel multi-modal distillation design for MTL, relying on pair-wise cross-task attentions mechanisms (coined xTAM, Section 3.1) combined into a multi-task exchange block (coined mTEB, Section 3.2.1). We address three critical tasks for outdoor scene understanding: semantic segmentation, dense depth estimation and surface normal estimation. In extensive experiments on multiple benchmarks, our proposed framework outperforms competitive MTL baselines (Section 4.2). With our new multi-task module, we also report improvement in semantic segmentation on Cityscapes where self-supervised depth plays as the auxiliary task [13]. Empirically, we showcase the merit of our proposed multi-task exchange in a new setting of MTL unsupervised domain adaptation (MTL-UDA, Section 4.5), outperforming all baselines.

## 2. Related Work

Multi-task learning. In the early days of neural networks, Caruana [3] introduces the idea of multi-task learning with *hard parameter sharing*, i.e. some parameters are shared between all tasks, some are dedicated to separate tasks. In the same spirit, UberNet [17] features a deep architecture to jointly address a large number of low-, mid- and high-level tasks. Zamir *et al.* [36] conduct a large-scale study with 26 tasks on four million images of indoor scenes, studying the dependencies and transferabilities across tasks. While those seminal works show great potential of multi-task learning, they do note a few challenges, most notably the *negative transfer* phenomenon that degrades performance of certain tasks when learned jointly [17, 15]. Some works [15, 5] reason that negative transfer is due to the imbalance of multi-task losses and introduce mechanisms to subsequently weight these individual loss terms. Kendall *et al.* [15] propose to weight multiple losses by estimating the homoscedastic uncertainty of each task. Chen *et al.* [5] introduce GradNorm, an algorithm that helps dynamically tuning gradient magnitudes such that the learning pace of different tasks are balanced. Differently, Sener and Koltun [24] cast multi-task learning as multi-objective optimization where the goal is to find Pareto-optimal solutions. Cross-task mechanisms. Closer to our work are methods focusing on improving exchange or distillation across tasks [33, 37, 27, 2]; the main idea is that each task could benefit from different yet complementary signals from the others. Inspired from the success of visual attention in perception tasks [34, 29, 8], PAD-Net [33] uses an attention mechanism to distill information across multi-modal features. In MTI-Net [27], Vandenhende *et al.* extend PAD-Net with a multi-scale solution to better distill multi-modal information. Zhang *et al.* [37] propose to aggregate affinity maps of all tasks, which is then “diffused” to refine task-

specific features. PSD [38] mines and propagates patch-wise affinities via graphlets, instead we model the interactions with attention mechanisms. Bruggemann *et al.* [2] introduce ATRC to enhance multi-modal distillation with four relational context types based on different types and levels of attention. Similar to ATRC, our method also exploits pairwise task similarity to refine task-specific features. Differently though, we advocate to combine the pairwise similarity with cross-task self-attention via a learnable weighting scheme to learn the refined residuals that complement the original task-specific features.

Recent efforts seek for label-efficient learning paradigms to train models for urban scene understanding. Tosi *et al.* [26] introduces real-time multi-task network based on knowledge distillation and self-supervision training. Hoyer *et al.* [13] propose a novel architecture and different strategies to improve semantic segmentation with self-supervised monocular depth estimation. One finding in [13] is that the MTL module of PAD-Net [33] complements other self-supervised strategies and further improve segmentation performance. On this regard, we study the effect brought by our proposed module in this particular setup.

## 3. Method

In multi-task learning, the aim is to optimize a set of  $n$  tasks:  $i \in \{T_1, \dots, T_n\}$  while seeking a *general* good performance – as opposed to favoring a single task. Our model takes an input image and makes  $n$  predictions. Generally, this is achieved with a shared encoder and separate task-specific decoders. Here, we introduce a novel mechanism that enhances cross-task talks via features exchange, building on the intuition that each decoder discovers unique but complementary features due to its separate supervision [36].

We formulate in Section 3.1 a bidirectional cross-Task Attention Mechanism module, coined xTAM, taking as input a pair of task-specific features and returning two *directional features*. We then present in Section 3.2 our complete MTL framework for scene understanding that encompasses our multi-Task Exchange Block (mTEB, see Fig. 1).

### 3.1. xTAM: Bidirectional Cross-Task Attention Mechanism

Building upon recent works demonstrating task interactions [25], our module seeks to capture the shared pairwise task knowledge while preserving their exclusive (non-shared) knowledge. Our intuition is that we can exploit features from pairs of tasks denoted  $(i, j)$  and self-discover the interactions from their correlation matrices for either directional interaction:  $i \rightarrow j$  and  $j \rightarrow i$ . Fig. 2 illustrates our xTAM component which distills the knowledge between tasks, taking as input the task features  $(f_i, f_j)$  and returning the *directional features* for the other task, i.e.  $(f_{j \rightarrow i}, f_{i \rightarrow j})$ .

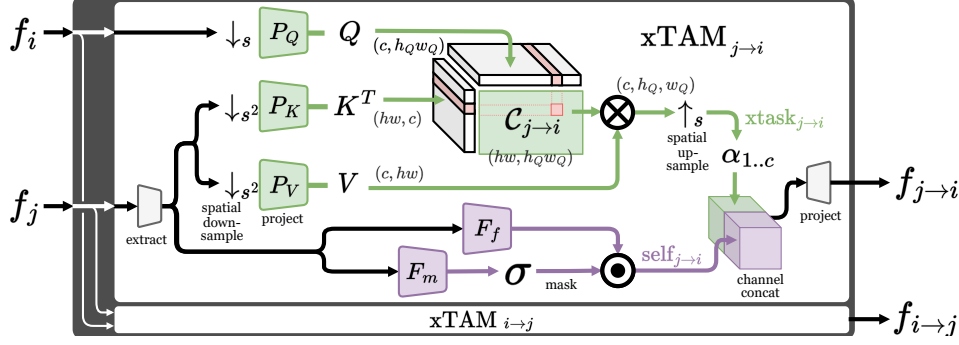


Figure 2: **Bidirectional Cross-task Attention (xTAM)**. We enable information flow between task pairs  $(i, j)$  via the discovery of directional features,  $f_{j \rightarrow i}$  and  $f_{i \rightarrow j}$ . Only  $\text{xTAM}_{j \rightarrow i}$  is detailed here. It relies on two attention mechanisms. First, a correlation-guided attention (green) to output  $\text{xtask}_{j \rightarrow i}$ , the features from task  $j$  contributing to task  $i$ . Second, a self-attention (purple) to discover complementary signal  $\text{self}_{j \rightarrow i}$  from  $j$ . Best viewed in color.

For ease of speech we describe the directional  $\text{xTAM}_{j \rightarrow i}$  where task  $j$  helps task  $i$ , but emphasize that xTAM being bidirectional, is made of both  $\text{xTAM}_{j \rightarrow i}$  and  $\text{xTAM}_{i \rightarrow j}$ .

### 3.1.1 Directional cross-task attention ( $\text{xTAM}_{j \rightarrow i}$ ).

Considering here  $i$  as our primary task, and  $j$  as the secondary, we seek to estimate the features from task  $j$  that can contribute to  $i$ . We leverage two cross-task attentions: (i) a *correlation-guided attention* which goal is to exploit cross-task spatial correlations to guide the extraction of contributing features of the secondary task to the primary task, and (ii) a *self-attention* on the secondary task to self-discover complementary signals which are beneficial for the primary task. Visualizations of the two attentions are colored in green and purple in Fig. 2, respectively. Note that each attention contributes differently to the primary task  $i$  relying either on identification of shared  $j$  and  $i$  knowledge or on exclusive  $j$  knowledge. We use  $1 \times 1$  convolutional layers (gray blocks in Fig. 2) for dimension compatibility.

**Correlation-guided attention.** To guide features, we rely on spatial correlations of task features (green blocks in Fig. 2). In practice, we project *downscaled*  $f_i$  and  $f_j$  features onto  $d$ -dimensional sub-spaces as in [29] such that  $K = P_Q(\downarrow_s f_i)$ ,  $Q = P_K(\downarrow_{s^2} f_j)$ , having  $\downarrow_s$  the downscale operator with  $s$  the scale factor, and  $P_Q, P_K$  separate  $1 \times 1$  convolutions. The spatial-correlation matrix  $C_{j \rightarrow i}$  is then obtained by applying a softmax on the matrix multiplication  $K^T \times Q$  and normalizing with  $\sqrt{d}$ :

$$C_{j \rightarrow i} = \text{softmax} \left( \frac{K^T \times Q}{\sqrt{d}} \right). \quad (1)$$

where  $d$  is the feature size [29]. Intuitively,  $C_{j \rightarrow i}$  has high values where features from  $i$  and  $j$  are highly correlated and low values otherwise, which we use to weight features from  $f_j$ . Subsequently, we obtain our correlation-guided

attention features  $\text{xtask}_{j \rightarrow i}$  by multiplying the correlation matrix  $C_{j \rightarrow i}$  with the projected  $j$  features:

$$\text{xtask}_{j \rightarrow i} = \uparrow_s (V \times C_{j \rightarrow i}), \quad (2)$$

with  $V = P_V(\downarrow_{s^2} f_j)$ ,  $P_V$  the  $1 \times 1$  projection, and  $\uparrow_s$  the upsample operator.

**Self-attention.** We additionally employ a spatial attention [33] which we denote as ‘self-attention’ to contrast with the above which takes pairs of differing tasks. Instead the following mechanism (purple blocks in Fig. 2) takes features from  $j$  alone and aims to extract private information from  $f_j$  that are relevant for predicting task  $i$ :

$$\text{self}_{j \rightarrow i} = F_f(f_j) \odot \sigma(F_m(f_j)), \quad (3)$$

being our *self-attention features*. Where  $\odot$  is the element-wise multiplication, and  $\sigma$  the sigmoid function. Both  $F_f$  and  $F_m$  are convolutional layers which are supervised by the target task  $i$  to learn to extract relevant information from features  $f_j$ . The self-attention features  $\text{self}_{j \rightarrow i}$  is defined as the point-wise multiplication between the features coming from  $F_f$  and the dynamic mask provided by  $F_m$ .

**Directional feature.** To construct the final *directional features*  $f_{j \rightarrow i}$  for the  $j \rightarrow i$  interaction, the two attention based feature maps are combined as:

$$f_{j \rightarrow i} = [\text{diag}(\alpha_1, \dots, \alpha_c) \times \text{xtask}_{j \rightarrow i}, \text{self}_{j \rightarrow i}], \quad (4)$$

where  $[\cdot, \cdot]$  is the channel-wise concatenation operation, and  $\alpha_{1..c}$  are learnable *scalars* used to weight the  $c$  channels of  $\text{xtask}_{j \rightarrow i}$ . All  $\alpha_{1..c}$  are initialized with 0; learning will adaptively adjust the per-channel weighting. Intuitively, cross-task exchange first starts with self-attention only, then gradually adjusts the  $\alpha_{1..c}$  values to include some contribution from the correlation-guided attention. This initialization strategy is important to stabilize training, especially here where we combine different types of attention.

Overall, the bidirectional xTAM block is made up of two directional blocks,  $\text{xTAM}_{j \rightarrow i}$  and  $\text{xTAM}_{i \rightarrow j}$ ; it outputs both  $f_{j \rightarrow i}$  and  $f_{i \rightarrow j}$  for each  $(i, j)$  task pair.

### 3.1.2 Discussion.

Our xTAM design is different from existing multi-modal distillation modules [33, 27, 2]. While PAD-Net [33] and MTI-Net [27] only consider cross-task self-attention to distill multi-modal information, ATRC [2] models pair-wise task similarity, which shares a similar spirit to our correlation-guided attention. Different to those, our xTAM learns to adaptively combine cross-task self-attention and correlation-guided attention using a learnable weighting scheme. Of note, adopting the multi-scale strategy of MTI-Net [27] or having other pair-wise modules of ATRC [2] is orthogonal to our xTAM. There might be potential improvements by systematically combining those strategies with xTAM. However, we focus here on giving an extensive study of xTAM design and applications; we leave such combinations for future investigation.

### 3.2. Multi-task Learning Framework.

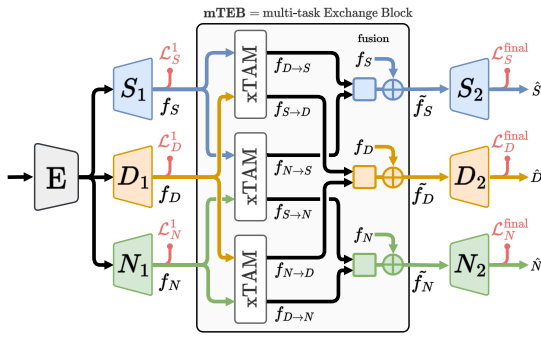


Figure 3: **Our cross-task MTL framework.** We jointly predict semantic segmentation ( $S$ , blue), depth ( $D$ , orange) and surface normal ( $N$ , green). All tasks share encoder ( $E$ ) and have dedicated decoders (shown as trapezoids). Our cross-task attention module is inserted within a “multi-Task Exchange Block” (mTEB) (Section 3.2.1) that takes task-specific features as inputs and returns refined features of the same kinds. Here, three xTAM modules are used to extract pairs of *directional features*, where each represents the complementary signal from one task to another, e.g.  $f_{D \rightarrow S}$  encapsulates depth information ( $D$ ) extracted to improve the segmentation task ( $S$ ). Multi-modal *directional features* are combined in the square blocks, resulting in multi-modal residuals for refinement. Task specific losses (shown in red) are applied before and after our mTEB.

We focus on semantic and geometrical estimation tasks as they are critical for scene understanding especially outdoor environments. Fig. 3 overviews our MTL framework having a unique encoder  $E$  and illustrates the interactions taking place between the three different task decoders ( $S$ ,  $D$  and  $N$ ) in a three-task setup:  $T = \{\text{Segmentation,}$

Depth, Normal $\}$ . We first describe our “multi-Task Exchange Block” (mTEB) and then go over our training setup.

#### 3.2.1 multi-Task Exchange Block (mTEB).

Our block can be inserted at any stage in the decoder, taking input task-specific features as inputs, here  $(f_S, f_D, f_N)$ , and outputting refined features, here  $(\tilde{f}_S, \tilde{f}_D, \tilde{f}_N)$ . The block consists of one bidirectional xTAM module (Section 3.1) per task pair, each returning two directional features which are fused to the input task features. Considering  $i$  to be the primary task, we concatenate the  $n-1$  *directional features*  $f_{j \rightarrow i} |_{j \in T \setminus \{i\}}$ , contributing towards task  $i$ , along the channel dimension. The concatenated feature map is then processed with  $F_{\rightarrow i}$  (depicted as square blocks in Fig. 3): a  $1 \times 1$  convolution followed by a batch-normalization [14] and ReLU activation. Projected features are then fused with the main task base features,  $f_i$ , via an element-wise addition denoted  $\oplus$ , leading to the final refined features  $\tilde{f}_i$ :

$$\tilde{f}_i = f_i \oplus F_{\rightarrow i}([f_{j \rightarrow i} |_{j \in T \setminus \{i\}}]), \quad (5)$$

where  $[...]$  is the channel-wise concatenation operation. Intuitively, the xTAM *directional features* are combined in  $F_{\rightarrow i}$  to constitute multi-modal residuals that complement original features  $f_i$ . By design, the refined features have same dimensions as the input task-specific features.

Although it is possible to insert a mTEB at several levels in the decoder. We show experimentally that the overall best choice is to have a single block before the last layer.

#### 3.2.2 Training.

To train our framework, we use a multi-scale multi-task objective defined as a linear combination of the task losses applied at two or more stages of the decoder: before any mTEB, and on the full resolution output. Although we do not explicitly introduce losses to encourage cross-task distillation, this is made possible implicitly by the design of xTAM and mTEB. More details in the supplementary.

### 4. Experiments

We present results and ablation studies of our proposed framework. While the dense multi-task learning literature lacks a unified setup for benchmarking, we strive to be comprehensive in our evaluation by including 4 tasks combined into 3 tasks sets, and evaluate on 4 datasets with 4 baselines – each requiring individual retraining. Section 4.1 describes our experimental and multi-task setup.

The evaluation of our contributions is threefold. First, in Section 4.2 our MTL proposal is evaluated on three task sets: ‘ $S$ - $D$ ’ (segmentation + depth), ‘ $S$ - $D$ - $N$ ’ (+ normals), and ‘ $S$ - $D$ - $N$ - $E$ ’ (+ edges). In Section 4.3, we ablate the position and choice of fusion in mTEB and alternative designs for the correlation-guided attention in xTAM.

Second, in Section 4.4 we demonstrate the ability of our multi-task strategy to maximize semantic segmentation performance when learning self-supervised depth regression.

Third, in Section 4.5 we further extend and evaluate our framework for MTL unsupervised domain adaptation.

## 4.1. Experimental setup

### 4.1.1 Common setup.

**Datasets.** We leverage two synthetic and two real datasets: VKITTI2 [9], Synthia [22], Cityscapes [6], and NYUDv2 [20]. In Cityscapes, we use depth from stereo via semi-global matching. Surface normals labels are estimated from depth maps following [35]. Lastly, NYUDv2 provides ground-truth semantic edge maps which we use for supervision. We train semantic segmentation on 14 classes on VKITTI2, 40 classes on NYUDv2, and 16 classes on Synthia and Cityscapes. In VKITTI2 and Synthia, we perform a random 80/20 split. On Cityscapes and NYUDv2, we use the official splits and load the images at  $1024 \times 512$  and  $576 \times 448$ , respectively. They are scaled to  $1024 \times 320$  on VKITTI2 and  $1024 \times 512$  on Synthia.

**Baselines.** We follow the recent survey [28] and compare our performance against single task learning networks (STL), where each task is predicted separately by a dedicated networks, and a naive multi-task learning baseline (MTL) using a shared encoder and task-specific decoders. PAD-Net [33] is a more competitive baseline than MTL. We address two variants of PAD-Net: (i) the original model used in [33] and (ii) a stronger model introduced in [13] (‘3-ways<sub>PAD-Net</sub>’). Seeking fair comparison, note that we applied a substantial experimental effort by best retraining and adapting each baseline to our setups. Details are in the supp.

**Architecture.** All networks have a shared ResNet-101 [12] encoder initialized with ImageNet [7], and a suitable number of decoders depending on the task set. Decoders for STL, MTL and PAD-Net baselines are from the survey [28]. While we stick to [13] for 3-ways<sub>PAD-Net</sub> and our model, using an *Atrous Spatial Pyramid Pooling* [4] followed by four upsampling blocks with skip connections [21].

**Training details.** We train all models for 40k iterations, set gradient norm clipping to 10 and reduce the learning rates by 0.1 at 30k steps as in [13]. We train our MTL models with the Adam optimizer [16] with  $\beta_1 = 0.9$  and  $\beta_2 = 0.98$ ; learning rate is set to  $2.0e-4$  for the backbone, and  $3.0e-4$  for all decoders. In other models, we use the SGD optimizer, setting the learning rates of the backbone and decoders to  $1.0e-3$  and  $1.0e-2$ , respectively, as in [13]. We use momentum of 0.9 and weight decay of  $5.0e-4$ .

We report the mean and standard deviation over three runs.

**Supervision.** We use the cross-entropy for semantic segmentation  $\mathcal{L}_S$  and edge estimation  $\mathcal{L}_E$  [2]. The berHu loss [18] is applied on inverse normalized values for depth regression as  $\mathcal{L}_D(\hat{d}) = \text{berHu}(d_{far}/\hat{d}, d_{far}/d)$ . To super-

vise surface normals training, we use the approach of [11]:  $\mathcal{L}_N(\hat{n}) = 1 - \cos(\hat{n}, n)$ , which computes the cosine between the estimated  $\hat{n}$  and ground-truth  $n$  normal vectors as  $\cos(\hat{n}, n) = \hat{n} \cdot n / (\|\hat{n}\| \|n\|)$ . Losses are weighted using our task balancing strategy.

**Single-task metrics.** We report a standard metric for each task: the mean intersection over union (mIoU) for semantic segmentation, the root mean square error in meters (RMSE) for depth regression, the mean error in degrees (mErr) for surface normals estimation, and the F1-score (F1) for semantic edge estimation.

### 4.1.2 Multi-task setup.

**Task balancing.** We observe that tasks loss weighting heavily impacts the performance of multi-task models similarly to [28], though we also notice that weights vary primarily with the loss functions and their interaction, and much less on the data or model. This is fairly reasonable knowing gradients scale with the loss term magnitude. Hence, as in [28] we applied grid-search for each unique task set, and use the same optimal set of weight for all methods. For  $T = \{S, D\}$  we obtain  $\{w_S=50, w_D=1\}$ , for  $T = \{S, D, N\}$  we get  $\{w_S=100, w_D=1, w_N=100\}$ , and choose for  $T = \{S, D, N, E\}$ ,  $\{w_S=100, w_D=1, w_N=100, w_E=50\}$ . Importantly, this search is done on the MTL baseline ensuring we do not favor our method in any way.

**MTL metrics.** Again, our goal is to improve the *overall* performance of the model over a set of tasks  $T$ , which we measure using the delta metric of [28] written  $\Delta_T$ . The latter measures the relative performance of a given multi-task model compared to the performance of dedicated Single Task Learning (STL) models:  $\Delta_T(\mathbf{f}) = 1/n \sum_{i \in T} (-1)^{g_i} (m_i - b_i)/b_i$ , where  $m_i$  and  $b_i$  are the metrics on task  $i$  of the model  $\mathbf{f}$  and STL model, respectively, and  $g_i$  is the metric direction of improvement, *i.e.*  $g_i=0$  when higher is better, and  $g_i=1$  when lower is better.

## 4.2. Main results

Table 1 (outdoor) and Table 2 (indoor) report our main experiments on 4 datasets, and 3 multi-task setups: ‘ $S-D$ ’, ‘ $S-D-N$ ’ and ‘ $S-D-N-E$ ’. In almost all setups our cross-task attention mechanism outperforms the baselines.

From Table 1, on the synthetic Synthia data with perfect segmentation and depth labels, our model outperforms all baselines by large margins of up to  $+5.69\Delta_{SD}$  in ‘ $S-D$ ’ and up to  $+5.75\Delta_{SDN}$  in ‘ $S-D-N$ ’. In real Cityscapes data, having stereo depth pseudo labels, our model is outperformed by 3-ways<sub>PAD-Net</sub> only in the ‘ $S-D$ ’ setup, and outperforms 3-ways<sub>PAD-Net</sub> in ‘ $S-D-N$ ’. The improvements of PAD-Net over MTL baselines confirm the benefit of self-attention mechanism [33]. Revisiting the “3-ways” architecture introduced in [13] in a different context of multi-task learning,

we observe a significant leap in performance of 3-ways<sub>PAD-Net</sub> over PAD-Net; the two models are only different in decoder design. This shows the importance of decoder in MTL.

Furthermore, the best overall performance (*i.e.* delta metrics) coincides with the best individual task performances, which shows the benefit of handling multiple tasks together. It is striking to look at the ‘*S-D-N*’ setup, where we also account for the rarely studied normal estimation task. In addition to the  $\Delta_{SDN}(f_{SDN})$  metric, we report a partial delta measurement  $\Delta_{SD}(f_{SDN})$ . This allows us to compare  $f_{SD}$  to  $f_{SDN}$ , respectively, the ‘*S-D*’ and ‘*S-D-N*’ models considering normal estimation as an auxiliary task. We record a gap of up to +3.23 points between  $\Delta_{SD}(f_{SDN})$  and  $\Delta_{SD}(f_{SD})$ , showing the benefit of injecting additional geometrical cues in the form of surface normals to help the other two tasks. We note that Ours follows this same observation, having  $\Delta_{SD}(f_{SD}) < \Delta_{SD}(f_{SDN})$  across all datasets.

In Table 2 we report indoor results on NYUDv2, which has additional Edge labels, and thus reporting ‘*S-D*’, ‘*S-D-N*’, and the newly formed ‘*S-D-N-E*’. Our method outperforms all baselines in the 3 task sets. The partial delta metrics  $\Delta_{SDN}(f_{SD})$  and  $\Delta_{SDN}(f_{SDNE})$  underline the advantage of normals and edges estimation, respectively. Using edge information for ‘Ours’ not only improved the partial  $\Delta_{SDN}$  in ‘*S-D-N-E*’ compared to ‘*S-D-N*’ (*i.e.*, +7.52 vs +6.49) but also led to better individual S/D/N metrics. This shows our framework benefits from additional cues for better scene understanding.

Figure 4 shows qualitative results on the Cityscapes dataset. This is noticeable when looking at thin elements (*e.g.* pedestrians, bicycles, *etc.*) and object contours. The visuals are in fair agreement with the quantitative analysis. More results are in the supplementary.

### 4.3. Ablation studies

We report ablations of our cross-task distillation module on the VKITTI2 dataset on both the ‘*S-D*’ and ‘*S-D-N*’ setups. We evaluate our method using different combinations of our module, varying the position and number of exchange blocks (Table 3), changing the type of fusion used (Table 4a), or the type of attention used (Table 4b).

**Multi-task exchange.** In Table 3 we report variations of our MTL framework with mTEB (cf. Section 3.2.1) positioned at different scales in the decoders, where scale  $s$  is the number of downsamplings from input resolution. We observe that with a single block (upper part, Table 3), the  $\Delta_{SD}$  and  $\Delta_{SDN}$  metrics are lower if the block is placed early in the decoder (*i.e.* scale 4) as opposed to being later in the decoder (*i.e.* scale 1). This follows the intuition that late features are more task-specific, thus better appropriate for cross-task distillation. Using several blocks (lower part, Table 3) was shown marginally beneficial, and at a cost of significantly more parameters (‘Param.’ column).

**Multi-task features fusion.** Considering a single mTEB at scale 1, we replace the original *add* fusion operator in Eq. (5) with either a *concatenation* or a *product* and report results in Table 4a showing that this choice impacts the overall delta metric. We notice that the gap differs on other datasets though ‘add’ and ‘prod’ remain always best.

**Correlation-guided attention.** While our original xTAM design (cf. Section 3.1) uses spatial attention with Eq. (1) for cross-task correlation-guided exchange, other practices exist in the literature [27]. Hence, in Table 4b we replace our choice of attention in Eq. (1) using either *spatial* attention, *channel* attention, or *both*. Results show that *channel* attentions or *both* are less efficient here, and come at higher implementation complexity.

In the supplementary material, we show it is preferred to combine correlation-guided attention and self-attention in xTAM in order to reach best performances.

### 4.4. MTL for segmentation

We now evaluate our multi-task strategy by focusing on semantic segmentation – often seen as a core task for scene analysis. In Table 5, we report performance training with two tasks (semantics+depth) on ‘Cityscapes SDE’ where unlike prior results we use Self-supervised monocular Depth Estimation (SDE) with Monodepth2 [10], as in “3-ways” [13]. In the latter, a proposed strategy is to use the multi-task module of PAD-Net (“3-ways<sub>PAD-Net</sub>” in Table 5). We proceed similarly but replacing the above module with our mTEB (hereafter, “3-ways<sub>mTEB</sub>”). We refer to [13] and the supplementary for in-depth technical details. Note that neither “3-ways<sub>PAD-Net</sub>” nor “3-ways<sub>mTEB</sub>” use the strategies of “3-ways” like data augmentation and selection for annotation. Still, Table 5 shows that using our module, *i.e.* 3-ways<sub>mTEB</sub>, achieves best results (+3.20 mIoU). This shows the increased benefit of our multi-task module.

### 4.5. MTL for UDA

Unsupervised domain adaptation (UDA) is the line of research dealing with distribution shifts between the *source* domain, in which we have labeled data, and the *target* domain, in which only unlabeled data is available for training. Here, we extend our experiments to the novel MTL Unsupervised Domain Adaptation setup (MTL-UDA) where the goal is to perform well *in average* on all tasks in the target domain. We argue that task exchange is beneficial for MTL-UDA since semantic- and geometry-related tasks exhibit different behaviors and sensitivities to the shift of domain, and are shown to be complementary.

We leverage typical synthetic to real scenarios: Synthia→Cityscapes and VKITTI2→Cityscapes, reporting results on the ‘*S-D*’ set. While semantic segmentation is degraded due to the shifts in colors, depth estimation is more affected by changes in scene composition and sensors.



Methods	'S-D'			'S-D-N'				
	Semseg $\uparrow$ mIoU %	Depth $\downarrow$ RMSE m	Delta $\uparrow$ $\Delta_{SD}$ %	Semseg $\uparrow$ mIoU %	Depth $\downarrow$ RMSE m	Delta $\uparrow$ $\Delta_{SD}$ %	Normals $\downarrow$ mErr. °	Delta $\uparrow$ $\Delta_{SDN}$ %
Synthia	STL [28]	67.43 $\pm 0.15$	5.379 $\pm 0.055$	$\downarrow$	<i>idem</i>	<i>idem</i>	19.61 $\pm 0.12$	$\downarrow$
	MTL [28]	69.83 $\pm 0.25$	5.166 $\pm 0.063$	+03.76 $\pm 0.77$	71.27 $\pm 0.21$	5.108 $\pm 0.076$	+05.37 $\pm 0.83$	+05.45 $\pm 0.72$
	PAD-Net [33]	70.87 $\pm 0.15$	4.917 $\pm 0.014$	+06.85 $\pm 0.24$	72.27 $\pm 0.25$	4.949 $\pm 0.072$	+07.58 $\pm 0.56$	+05.62 $\pm 0.43$
	3-ways <sub>PAD-Net</sub> [13]	77.50 $\pm 0.17$	4.289 $\pm 0.028$	+17.60 $\pm 0.13$	79.93 $\pm 0.5$	4.218 $\pm 0.082$	+20.06 $\pm 0.92$	+20.29 $\pm 0.84$
	Ours	<b>80.53</b> $\pm 0.43$	<b>4.161</b> $\pm 0.022$	+21.04 $\pm 0.52$	<b>82.99</b> $\pm 0.38$	<b>4.056</b> $\pm 0.076$	+23.83 $\pm 0.98$	+24.92 $\pm 0.87$
VKIT12	STL [28]	84.53 $\pm 0.06$	5.720 $\pm 0.027$	$\downarrow$	<i>idem</i>	<i>idem</i>	23.14 $\pm 0.68$	$\downarrow$
	MTL [28]	87.73 $\pm 0.12$	5.720 $\pm 0.029$	+01.89 $\pm 0.21$	87.83 $\pm 0.21$	5.714 $\pm 0.033$	+02.00 $\pm 0.27$	+02.54 $\pm 0.80$
	PAD-Net [33]	88.43 $\pm 0.12$	5.571 $\pm 0.058$	+03.63 $\pm 0.45$	88.67 $\pm 0.15$	5.543 $\pm 0.043$	+04.09 $\pm 0.29$	+04.09 $\pm 0.83$
	3-ways <sub>PAD-Net</sub> [13]	96.13 $\pm 0.15$	4.013 $\pm 0.051$	+21.78 $\pm 0.54$	96.87 $\pm 0.06$	3.756 $\pm 0.013$	+24.46 $\pm 0.14$	+27.25 $\pm 0.90$
	Ours	<b>97.00</b> $\pm 0.10$	<b>3.423</b> $\pm 0.025$	+27.47 $\pm 0.16$	<b>97.53</b> $\pm 0.06$	<b>3.089</b> $\pm 0.006$	+30.70 $\pm 0.05$	+33.00 $\pm 0.73$
Cityscapes	STL [28]	67.93 $\pm 0.06$	6.622 $\pm 0.020$	$\downarrow$	<i>idem</i>	<i>idem</i>	44.10 $\pm 0.01$	$\downarrow$
	MTL [28]	70.43 $\pm 0.12$	6.797 $\pm 0.520$	+00.52 $\pm 0.32$	70.93 $\pm 0.15$	6.736 $\pm 0.023$	+01.34 $\pm 0.28$	+01.30 $\pm 0.18$
	PAD-Net [33]	70.23 $\pm 0.25$	6.777 $\pm 0.010$	+00.52 $\pm 0.27$	70.67 $\pm 0.06$	6.755 $\pm 0.018$	+01.00 $\pm 0.17$	+01.12 $\pm 0.11$
	3-ways <sub>PAD-Net</sub> [13]	<b>75.00</b> $\pm 0.10$	<b>6.528</b> $\pm 0.063$	+05.91 $\pm 0.44$	<b>75.50</b> $\pm 0.10$	<b>6.491</b> $\pm 0.081$	+06.56 $\pm 0.61$	+06.09 $\pm 0.37$
	Ours	<b>74.95</b> $\pm 0.10$	<b>6.649</b> $\pm 0.003$	+04.96 $\pm 0.08$	<b>76.08</b> $\pm 0.14$	<b>6.407</b> $\pm 0.013$	+07.61 $\pm 0.04$	+08.15 $\pm 0.22$

Table 1: **MTL performance on two sets.** We report individual task metrics but seek best overall performance measured by  $\Delta_{SD}$  and  $\Delta_{SDN}$ , computed w.r.t. “STL”. Except for ‘S-D’ on Cityscapes where we are second, we outperform significantly the baselines on all delta metrics. We also report  $\Delta_{SD}$  in ‘S-D-N’ for direct comparison with ‘S-D’. Notice  $\Delta_{SD}(f_{SDN}) > \Delta_{SD}(f_{SD})$  across methods highlighting the importance of surface normals estimation. We highlight **best** and **2nd best**.

Methods	'S-D'			'S-D-N'					'S-D-N-E'					
	Semseg $\uparrow$ mIoU %	Depth $\downarrow$ RMSE m	Delta $\uparrow$ $\Delta_{SD}$ %	Semseg $\uparrow$ mIoU %	Depth $\downarrow$ RMSE m	Delta $\uparrow$ $\Delta_{SD}$ %	Normals $\downarrow$ mErr. °	Delta $\uparrow$ $\Delta_{SDN}$ %	Semseg $\uparrow$ mIoU %	Depth $\downarrow$ RMSE m	Normals $\downarrow$ mErr. °	Delta $\uparrow$ $\Delta_{SDN}$ %	Edges $\uparrow$ F1 %	Delta $\uparrow$ $\Delta_{SDNE}$ %
STL [28]	38.70 $\pm 0.10$	0.635 $\pm 0.013$	$\downarrow$	<i>idem</i>	<i>idem</i>	$\downarrow$	36.90 $\pm 0.26$	$\downarrow$	<i>idem</i>	<i>idem</i>	<i>idem</i>	$\downarrow$	54.90 $\pm 0.00$	$\downarrow$
MTL [28]	<u>39.44</u> $\pm 0.34$	0.638 $\pm 0.004$	+1.63 $\pm 0.37$	39.90 $\pm 0.41$	0.642 $\pm 0.003$	+1.89 $\pm 0.67$	36.07 $\pm 0.09$	+1.76 $\pm 0.53$	39.70 $\pm 0.35$	0.636 $\pm 0.001$	36.10 $\pm 0.12$	+1.88 $\pm 0.33$	55.11 $\pm 0.15$	+1.50 $\pm 0.20$
PAD-Net [33]	35.30 $\pm 0.84$	0.659 $\pm 0.004$	-5.36 $\pm 0.83$	36.14 $\pm 0.30$	0.660 $\pm 0.006$	-4.32 $\pm 0.69$	36.72 $\pm 0.08$	-2.97 $\pm 0.43$	36.19 $\pm 0.24$	0.662 $\pm 0.005$	36.58 $\pm 0.06$	-2.92 $\pm 0.37$	54.79 $\pm 0.07$	-2.24 $\pm 0.26$
3-ways <sub>PAD-Net</sub> [13]	<b>39.47</b> $\pm 0.16$	<u>0.622</u> $\pm 0.001$	+2.90 $\pm 0.24$	<b>40.28</b> $\pm 0.30$	<u>0.619</u> $\pm 0.004$	+4.16 $\pm 0.50$	<b>35.35</b> $\pm 0.09$	+3.93 $\pm 0.27$	<u>40.16</u> $\pm 0.28$	<u>0.614</u> $\pm 0.000$	<b>35.25</b> $\pm 0.09$	+4.14 $\pm 0.65$	<b>59.66</b> $\pm 0.16$	+5.27 $\pm 0.49$
Ours	38.93 $\pm 0.35$	<b>0.604</b> $\pm 0.004$	+3.54 $\pm 0.31$	<b>40.28</b> $\pm 0.41$	<b>0.598</b> $\pm 0.002$	+5.80 $\pm 0.05$	<b>33.72</b> $\pm 0.14$	+6.49 $\pm 0.36$	<b>40.84</b> $\pm 0.37$	<b>0.593</b> $\pm 0.004$	<b>33.38</b> $\pm 0.19$	+7.52 $\pm 0.27$	<b>61.12</b> $\pm 0.24$	+8.47 $\pm 0.12$

Table 2: **Results on NYUDv2 evaluated on three sets of tasks.** We report a delta metric for each set as in Table 1 and provide partial metrics  $\Delta_{SD}(f_{SDN})$  and  $\Delta_{SDN}(f_{SDNE})$  to compare between sets and highlight the benefit of inserting additional tasks in the framework. We underline the fact that  $\Delta_{SDN}(f_{SDNE}) > \Delta_{SDN}(f_{SDN})$  validating the benefit of edge estimation even for the other tasks.

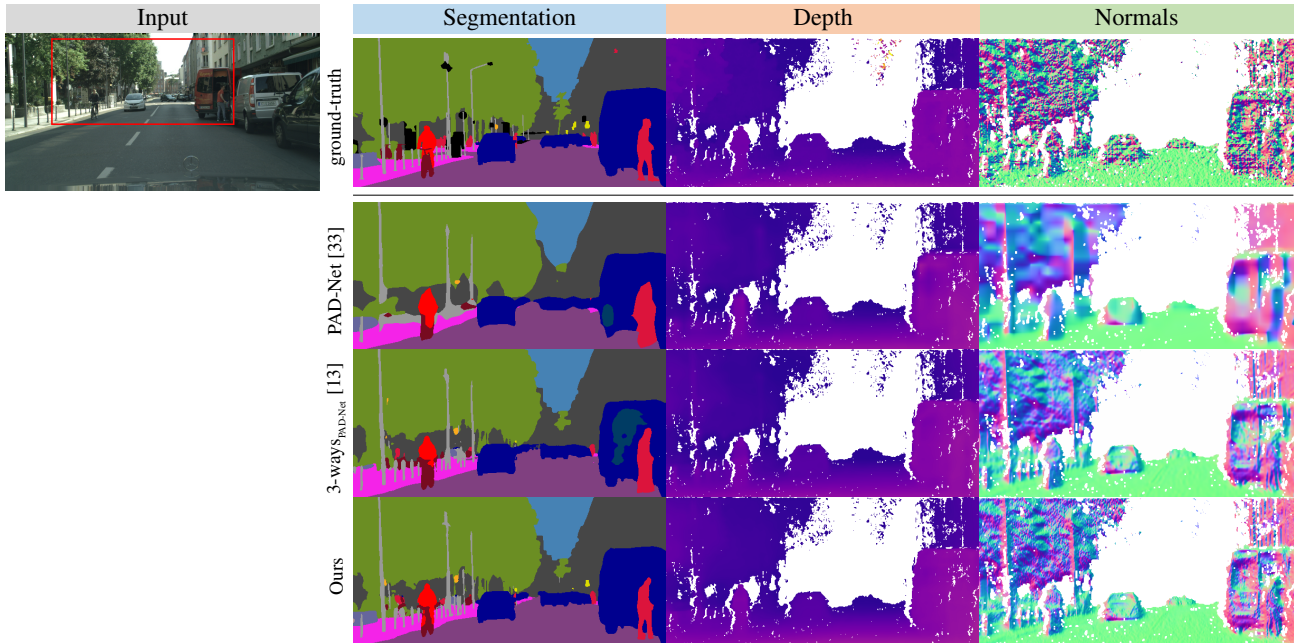


Figure 4: **Qualitative MTL results on Cityscapes in the ‘S-D-N’ setup.** The first row shows the input image and its segmentation, depth and normal ground-truths. Segmentation results of our models are better overall, especially in boundary areas – see “bicycle”, “rider” and “pedestrian”. Surface normals of PAD-Net is blurrier as compared to 3-ways<sub>PAD-Net</sub> and Ours.

mTEB Scales 4 3 2 1	'S-D'				'S-D-N'			
	Param. ↓	Semseg ↑	Depth ↓	Delta ↑	Param. ↓	Semseg ↑	Depth ↓	Delta ↑
	#M added	mIoU %	RMSE m	$\Delta_{SD}$ %	#M added	mIoU %	RMSE m	$\Delta_{SDN}$ %
	<b>0.00</b>	96.88 $\pm 0.30$	3.604 $\pm 0.020$	25.81 $\pm 0.33$	<b>0.00</b>	97.38 $\pm 0.02$	3.491 $\pm 0.041$	14.50 $\pm 0.57$
✓	3.09	<b>97.32</b> $\pm 0.06$	3.556 $\pm 0.029$	26.50 $\pm 0.29$	2.32	97.43 $\pm 0.04$	3.559 $\pm 0.024$	14.51 $\pm 0.50$
✓	3.09	97.24 $\pm 0.03$	3.476 $\pm 0.018$	27.15 $\pm 0.16$	2.32	97.49 $\pm 0.08$	3.353 $\pm 0.025$	14.45 $\pm 0.48$
✓	<b>0.77</b>	97.07 $\pm 0.06$	3.468 $\pm 0.016$	27.11 $\pm 0.12$	<b>9.26</b>	97.47 $\pm 0.06$	3.244 $\pm 0.035$	14.57 $\pm 0.51$
✓	<b>0.77</b>	97.00 $\pm 0.10$	<b>3.423</b> $\pm 0.025$	<b>27.47</b> $\pm 0.16$	<b>9.26</b>	<b>97.53</b> $\pm 0.06$	<b>3.089</b> $\pm 0.006$	<b>14.44</b> $\pm 0.52$
✓	<b>3.86</b>	<b>97.09</b> $\pm 0.03$	<b>3.369</b> $\pm 0.022$	<b>27.99</b> $\pm 0.18$	<b>11.58</b>	<b>97.53</b> $\pm 0.02$	<b>3.080</b> $\pm 0.025$	<b>14.47</b> $\pm 0.57$
✓	4.63	97.01 $\pm 0.02$	3.377 $\pm 0.008$	27.88 $\pm 0.06$	13.89	97.39 $\pm 0.02$	3.136 $\pm 0.046$	14.81 $\pm 0.77$
✓	7.72	97.05 $\pm 0.03$	<b>3.369</b> $\pm 0.010$	27.97 $\pm 0.08$	23.15	96.82 $\pm 0.23$	3.307 $\pm 0.066$	15.39 $\pm 0.65$

Table 3: **Ablations on VKITTI2 changing placement of mTEB.** The upper part shows our block is more efficient when located later in the decoders (*i.e.* scale 1) which we attribute to more task-specific features. Using more blocks leads to a small boost at the cost of more parameters ('Param.' column represents the size of distillation block).

Ablation	'S-D-N'			
	Semseg ↑ mIoU %	Depth ↓ RMSE m	Normals ↓ mErr. °	Delta ↑ $\Delta_{SDN}$ %
concat	97.43 $\pm 0.06$	3.315 $\pm 0.099$	14.83 $\pm 0.43$	31.08 $\pm 0.61$
prod	<b>97.63</b> $\pm 0.06$	<b>3.139</b> $\pm 0.024$	<b>14.34</b> $\pm 0.49$	<b>32.90</b> $\pm 0.81$
add	97.53 $\pm 0.06$	<b>3.089</b> $\pm 0.006$	<b>14.44</b> $\pm 0.52$	<b>33.00</b> $\pm 0.73$

(a) Fusion operation

spatial	<b>97.43</b> $\pm 0.06$	<b>3.315</b> $\pm 0.099$	<b>14.83</b> $\pm 0.43$	<b>31.08</b> $\pm 0.61$
channel	97.08 $\pm 0.24$	3.585 $\pm 0.047$	15.03 $\pm 0.51$	28.70 $\pm 0.92$
both	97.33 $\pm 0.11$	<b>3.352</b> $\pm 0.068$	<b>14.85</b> $\pm 0.48$	<b>30.42</b> $\pm 0.51$

(b) Type of attention

Table 4: **Ablation on VKITTI2.** We compare different fusion operators for Eq. (5), in 4a and different choices of attention for xTAM, in 4b.

Methods	road	swalk	build.	wall	fence	pole	light	sign	veg	sky	person	rider	car	bus	mbike	bike	mIoU %
3-ways [13]	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	71.16
3-ways <sub>PAD-Net</sub> [13]	97.21	79.38	90.50	<b>47.68</b>	49.68	51.17	49.41	64.65	91.40	93.85	72.41	46.92	92.66	80.17	42.43	66.39	69.74
3-ways <sub>mTEB</sub> ours	<b>97.62</b>	<b>82.29</b>	<b>92.44</b>	46.52	<b>54.76</b>	<b>59.82</b>	<b>60.94</b>	<b>73.13</b>	<b>92.22</b>	<b>94.55</b>	<b>76.40</b>	<b>58.49</b>	<b>94.26</b>	<b>85.14</b>	<b>49.41</b>	<b>71.70</b>	<b>74.36</b>

Table 5: **Semantic segmentation with SDE on Cityscapes.** We insert PAD-Net distillation block or mTEB: "3-ways<sub>PAD-Net</sub>" or "3-ways<sub>mTEB</sub>", respectively, in the architecture of [13]. \* We only report mIoU because [13] does not release their model weights trained on the full set.

**Architecture adjustments.** We adopt a simple multi-task Domain Adaptation (DA) solution based on output-level DA adversarial training [30, 31, 23]. As output-level techniques do not alter the base MTL architecture, it allows direct understanding whether our MTL design is favorable or not for adaptation. For each task, a small discriminator network is jointly trained with the main MTL model, which act as two sides in an adversarial game. While discriminators try to tell from which domain the input data comes from, the MTL model tries to fool all discriminators by making outputs of source and target domain indistinguishable, which eventually helps align source/target. The complete network and training are detailed in the supplementary materials.

**Baselines adjustments.** We use the same baselines as in Section 4.2 and introduce patch discriminators to align the processed output prediction maps. We use the STL-UDA models - consisting of a single task decoder and one output type - as baseline for measuring the delta metric. We train all methods on source and target data in mix batches containing two instances from either domains.

**Results.** In Table 6, results show that the use of our multi-task exchange significantly improves performance in all scenarios and metrics. Along with MTL results, we report STL *source*, trained only on source, and STL *oracle*, trained on labeled target. In Synthia→Cityscapes our method very significantly outperforms the naive MTL-UDA baseline by +38.1 $\Delta_{SD}$ , and by +8.34 $\Delta_{SD}$  in VKITTI2→Cityscapes, also coinciding with the best individual task metrics. An interesting note is that STL-UDA is preferable over straightforward multi-task learning in UDA (MTL-UDA), which we attribute to the competing objectives in MTL-UDA.

Methods	Synthia→Cityscapes (16 classes)			VKITTI2→Cityscapes (8 classes)		
	Semseg ↑ mIoU %	Depth ↓ RMSE m	Delta ↑ $\Delta_{SD}$ %	Semseg ↑ mIoU %	Depth ↓ RMSE m	Delta ↑ $\Delta_{SD}$ %
STL <i>target</i>	67.93 $\pm 0.06$	06.62 $\pm 0.02$	-	77.10 $\pm 0.10$	06.62 $\pm 0.02$	-
STL <i>source</i>	35.63 $\pm 0.67$	13.79 $\pm 0.28$	-	58.77 $\pm 0.06$	11.99 $\pm 0.34$	-
MTL <i>target</i>	70.43 $\pm 0.12$	06.79 $\pm 0.52$	-	79.63 $\pm 0.07$	06.72 $\pm 0.02$	-
STL-UDA	37.55 $\pm 0.92$	14.26 $\pm 0.14$	↘	61.60 $\pm 0.35$	11.45 $\pm 0.32$	↘
MTL <i>source</i>	15.32 $\pm 1.02$	14.51 $\pm 0.42$	-30.49 $\pm 2.58$	49.50 $\pm 1.64$	12.26 $\pm 0.69$	-13.37 $\pm 2.67$
MTL-UDA	16.71 $\pm 1.22$	14.47 $\pm 0.86$	-28.49 $\pm 2.41$	57.26 $\pm 0.71$	11.85 $\pm 0.32$	-05.31 $\pm 0.82$
Ours	<b>37.93</b> $\pm 0.79$	<b>11.66</b> $\pm 0.55$	<b>09.61</b> $\pm 0.87$	<b>63.76</b> $\pm 0.91$	<b>11.15</b> $\pm 0.08$	<b>03.03</b> $\pm 0.92$

Table 6: **MTL-UDA.** We consider the 'S-D' task set and report  $\Delta_{SD}$  metric w.r.t. single-task learning unsupervised domain adaptation (STL-UDA). Overall, naive multi-task adaptation strategy (MTL-UDA) is lowering performance w.r.t. STL-UDA, while our method outperforms all others

## 5. Conclusions

We address semantic and geometry scene understanding with a multi-task learning framework on the four tasks: semantic segmentation, dense depth regression, surface normal estimation, and edge estimation. We propose a novel distillation module, multi-Task Exchange Block, built upon xTAM, a bidirectional cross-task attention mechanism that combines two types of cross-task attention to refine and enhance all task-specific features. Extensive experiments in various datasets demonstrate the effectiveness of our proposed framework over competitive MTL baselines. We extend our work to two scenarios: (i) to improve semantic segmentation on a recent self-supervised framework [13] and (ii) to improve the performance of MTL-UDA.

**Acknowledgements.** This work was funded by the French Agence Nationale de la Recherche (ANR) with project SIGHT (ANR-20-CE23-0016) performed with HPC resources from GENCI-IDRIS (Grant 2021-AD011012808).



## References

- [1] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*. Physica-Verlag HD, 2010.
- [2] David Bruggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. Exploring relational context for multi-task dense prediction. In *ICCV*, 2021.
- [3] Rich Caruana. Multitask learning. *Machine learning*, 1997.
- [4] 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. *TPAMI*, 2018.
- [5] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, 2018.
- [6] 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 *CVPR*, 2016.
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.
- [8] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019.
- [9] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016.
- [10] Clément Godard, Oisín Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019.
- [11] Vitor Guizilini, Jie Li, Rares Ambrus, and Adrien Gaidon. Geometric unsupervised domain adaptation for semantic segmentation. In *ICCV*, 2021.
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [13] Lukas Hoyer, Dengxin Dai, Yuhua Chen, Adrian Köring, Suman Saha, and Luc Van Gool. Three ways to improve semantic segmentation with self-supervised depth estimation. In *CVPR*, 2021.
- [14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.
- [15] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018.
- [16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [17] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017.
- [18] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks, 2016.
- [19] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *CVPR*, 2015.
- [20] Pushmeet Kohli, Nathan Silberman, Derek Hoiem, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.
- [21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.
- [22] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016.
- [23] 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*, 2021.
- [24] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *NeurIPS*, 2018.
- [25] Trevor Standley, Amir R. Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *ICML*, 2020.
- [26] Fabio Tosi, Filippo Aleotti, Pierluigi Zama Ramirez, Matteo Poggi, Samuele Salti, Luigi Di Stefano, and Stefano Mattochia. Distilled semantics for comprehensive scene understanding from videos. In *CVPR*, 2020.
- [27] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *ECCV*, 2020.
- [28] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *TPAMI*, 2021.
- [29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [30] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, 2019.
- [31] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Dada: Depth-aware domain adaptation in semantic segmentation. In *ICCV*, 2019.
- [32] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [33] 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*, 2018.
- [34] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

- [35] Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. Unsupervised learning of geometry with edge-aware depth-normal consistency, 2017.
- [36] Amir Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In CVPR, 2018.
- [37] 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, 2019.
- [38] Ling Zhou, Zhen Cui, Chunyan Xu, Zhenyu Zhang, Chaoqun Wang, Tong Zhang, and Jian Yang. Pattern-structure diffusion for multi-task learning. In CVPR, 2020.