Learning Good Features to Transfer Across Tasks and Domains

Learning Good Features to Transfer Across Tasks and Domains

Pierluigi Zama Ramirez*[1], Adriano Cardace*[1], Luca De Luigi*[1],
Alessio Tonioni[2], Samuele Salti[1], Luigi Di Stefano[1]
[1]University of Bologna, Italy
[2]Google Inc.
{pierluigi.zama,adriano.cardace2,luca.deluigi4,samuele.salti,luigi.distefano}@unibo.it
alessiot@google.com

Abstract—Availability of labelled data is the major obstacle to the deployment of deep learning algorithms for computer vision tasks in new domains. The fact that many frameworks adopted to solve different tasks share the same architecture suggests that there should be a way of reusing the knowledge learned in a specific setting to solve novel tasks with limited or no additional supervision. In this work, we first show that such knowledge can be shared across tasks by learning a mapping between task-specific deep features in a given domain. Then, we show that this mapping function, implemented by a neural network, is able to generalize to novel unseen domains. Besides, we propose a set of strategies to constrain the learned feature spaces, to ease learning and increase the generalization capability of the mapping network, thereby considerably improving the final performance of our framework. Our proposal obtains compelling results in challenging synthetic-to-real adaptation scenarios by transferring knowledge between monocular depth estimation and semantic segmentation tasks.

Index Terms—Domain Adaptation, Task Transfer, Semantic Segmentation, Depth estimation

1 Introduction

Deep learning has revolutionized computer vision by providing an effective solution to address a wide range of tasks (e.g., classification, depth estimation, semantic segmentation, etc.). The rise of a common framework has allowed incredible leaps forward for the whole research community thanks to the ability to reuse architectural and algorithmic improvements discovered to solve one task across many others. However, the real knowledge of a neural network is stored inside its trained parameters and we still have no simple way of sharing this knowledge across different tasks and domains (i.e., datasets). As such, the first step for every practitioner faced with a new problem or domain deals with acquisition and labeling of a new training set, an extremely tedious, expensive and time consuming operation. We argue that sharing the knowledge acquired by a neural network to solve a specific task in a specific domain across other tasks and domains could be a more straightforward and cost-effective way to tackle them.

Indeed, this is demonstrated by the widespread use and success of *transfer learning*. Transfer learning concerns solving new tasks by initializing a network with pre-trained weights, thereby providing a basic approach to knowledge reuse. However, it still requires a new annotated dataset to fine tune the pretrained network on the the task at hand. A few works focused on the related *task transfer* (TT) problem [1], [2], i.e., on exploiting supervised data to tackle multiple tasks in a single domain more effectively by leveraging on the relationships between the learned representations. As unlabeled domains are not considered in TT problem formulations, the proposed methodologies still rely on transfer learning and availability of a small annotated training set in order to address new datasets. On the other hand, the unsupervised *domain adaptation* literature (DA) [3] studies how the need for annotated data can be removed when leveraging on knowledge reuse to solve the same task across different domains, but it does not consider different tasks.

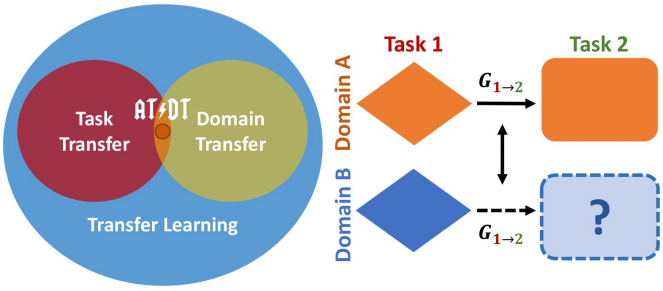


Fig. 1. Our framework transfers knowledge across tasks and domains. Given two tasks (1 and 2) and two domains (A and B), with supervision for both tasks in A but only for one task in B, we learn the dependency between the tasks in A and exploit this in B in order to solve task 2 without the need of supervision.

Differently, we propose to merge DA and TT by explicitly addressing a cross domain and cross task problem where on one source domain (e.g., synthetic data) we have supervision for many tasks, while in another target one (e.g., real data) annotations are available only for a specific task while we wish to solve many. A schematic representation of our problem formulation with two domains and two tasks is shown in the right part of Figure 1. Following this schematic representation we will consider a scenario with

* *Equal contribution.*

two domains (a source one and a target one, namely A and B) and two tasks (again a source one and a target one, namely task 1 and 2), but nothing prevents our method to be extended to more. In domain A we use the available supervision to learn two models for the source and target tasks, while in the target domain B we can do the same for the source task only. In domain A we use the trained task-specific models to learn a mapping function ($G_{1\rightarrow 2}$ in Figure 1) between deep features extracted to solve the source task and those extracted to solve the target task. This mapping function is then applied in domain B to solve the target task by transforming the features extracted to solve the source task.

The key component of our framework is the mapping function between the two task-specific deep features. In [4] we proposed a preliminary formulation of our framework by modeling the mapping function as a deep convolutional neural network and optimizing its parameters by standard supervised learning in the source domain A. In this work, we expand and improve upon our preliminary formulation by proposing two features alignment strategies aimed at learning the feature mapping function more effectively. Firstly, we align feature representations across domains using a novel norm discrepancy alignment (NDA) loss that constraints the feature space by penalizing features with very different norms in a spatially-aware manner. Secondly, we align feature representations across tasks by using them as inputs to solve a common auxiliary task. This pretext problem acts as a bridge between the source and the target tasks: in fact, if the deep features extracted to solve them independently can be used to address effectively an additional common task, we are pushed to believe that those features present the same semantic content and encode it in a similar manner.

We test the effectiveness of our proposal in a challenging autonomous driving scenario where we try to solve the two related dense prediction tasks of monocular depth estimation and semantic segmentation [5]. We select edge detection as the auxiliary task since color edges provide oftentimes detailed key information related to both the semantic as well as the depth structure of the scene. Many edge detectors have been proposed during the years, with recent deep learning based approaches outperforming classical handcrafted methods even in the most challenging scenarios [6], [7], [8]. Interestingly, such deep models present good generalization capabilities, allowing us to use the state-of-the-art approach [6] to generate proxy supervision for the auxiliary task without extra labels. Thanks to our formulation, we can use a fully supervised and completely synthetic domain (i.e., the Carla simulator [9]) to improve the performance on a partially labeled real domain (i.e., Cityscapes [10]).

The contributions of this paper can be summarized as follow:

- We propose for the first time to study a cross domain and cross task problem where supervision for all tasks is available in one domain whilst only for a subset of them in the other. This is done by learning a mapping between deep representations.
- We demonstrate how constraining explicitly deep features across domains with a novel norm discrepancy alignment loss improves the learning of the mapping function.
- We further show how the learning of the mapping function can be improved by deploying an auxiliary task.
- Considering the dense prediction tasks of monocular depth estimation and semantic segmentation, we achieve results close to the practical upper bound when transferring knowledge between a synthetic and a real domain.

2 Related Works

2.1 Transfer Learning and Task Transfer

Collecting training data is often expensive, time-consuming, or even unrealistic in many scenarios. Many works have tackled this problem by exploiting the existence of a relationship between the weights of CNNs trained for different tasks [11]. In particular, [12] showed that this strategy, referred to as transfer learning, can lead to better results than using random initialization even if applied on quite diverse tasks. Transfer learning has become a common practice, for instance, in object detection, where networks are usually initialized with Imagenet [13] classification weights [14], [15], [16], [17]. Additional insights on the transferability of learned representations between different visual tasks were provided in [1], where the authors present Taskonomy, a computational approach to represent a taxonomy of relationships among visual tasks. Along similar lines, [18] proposed to exploit the correlation between known supervised tasks and novel target tasks, in order to predict the parameters of models deployed to solve the target tasks starting from the parameters of networks trained on the known tasks. While [1] and [18] study the correlation between tasks in a given domain and assume either full or no supervision, we explicitly address a multi-domain scenario assuming full supervision in one domain and partial supervision in the target one.

2.2 Domain Adaptation

Domain adaptation techniques aim at reducing the performance drop of a model deployed on a domain different from the one the model was trained on [3]. Throughout the years, adaptation has been performed at different levels. Early approaches tried to model a shared feature space relying on statistical metrics such as MMD [19], [20]. Later, some works proposed to align domains by adversarial training [21], [22], [23]. Recently [24] noticed that, for classification tasks, aligning feature norms to an arbitrarily large value results in better transferability across domains. Generative adversarial networks [25] have also been employed to perform image-to-image translation between different domains [26], [27], [28], and, in particular, to render cheaply labelled synthetic images similar to real images from a target domain. However, when dealing with dense tasks such as semantic segmentation, feature-based domain adaptation approaches tend to fail as deeply discussed in [29] . Thus, several approaches to address domain adaptation for dense tasks, such as semantic segmentation [5], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40] or depth estimation [41], [42],

[43] have been proposed recently. Among them, SPIGAN [44] uses extra supervision coming from synthetic depth of the source domain to improve the quality of an image-to-image translation network and consequently achieving better adaptation performances. Akin to DA methods, we learn from a labeled source domain to perform well on a different target domain. However, unlike the classical DA setting, we assume the existence of an additional task where supervision is available for both domains.

2.3 Multi-task Learning

The goal of multi-task learning is to solve many tasks simultaneously. By pursuing this rather than solving the tasks independently, a neural network may use more information to obtain more robust and reliable predictions. Many works try to tackle several tasks jointly [45], [46], [47], [48]. For example, [47] showed that by learning to correctly weigh each task loss, multi-task learning methods can outperform separate models trained individually. [5], [48] show how learning multiple perception tasks jointly while enforcing geometrical consistency across them can lead to better performances for almost all tasks. Recently, [2] proposes a method to improve the performances of multiple single-task networks by imposing consistency across them during training. Finally, Taskonomy [1] investigates the relationship between the deployed tasks to accomplish multi-task learning effectively. However, multi-task learning approaches usually try to achieve the best balance between tasks in a single-domain scenario. We instead tackle a multi-task and multi-domain problem. Nevertheless, taking inspiration from multi-task learning, we show how jointly learning an auxiliary task while learning the two task networks helps the alignment of features across tasks.

2.4 Task Transfer and Domain Adaptation

Most existing approaches address independently either task transfer or domain adaptation. Yet, a few works have proposed to tackle these two problems jointly. [49] was the first paper to propose a cross-tasks and cross-domains adaptation approach, considering as tasks different image classifications problems. UM-Adapt [50], instead, learns a cross-task distillation framework with full supervision on the source domain and deploys such framework on the target domain in a fully unsupervised manner, while minimizing adversarially the discrepancy between the two domains. Differently, in a preliminary version of this work [4], we introduced AT/DT (Across Tasks and Domains Transfer) and set forth a novel learning framework, where the relationship between a set of tasks is learned on the source domain and it is later deployed to solve a specific task on the target domain without supervision thanks to the availability of ground-truth for all the tasks except the target one. In this work we will expand and improve this methodology.

3 Method

We introduce the problem we are trying to solve with a practical example. Imagine we aim to solve semantic segmentation in a real domain but we only have labels for a closely related task (e.g., depth estimation). Moreover, let us suppose to have access to a synthetic domain, where labels can be easily obtained for both tasks. Unsupervised domain adaptation may be used in this synthetic to real scenario. However, we wish to go one step further, trying to answer this question: can we exploit the depth estimation task to boost the performance of semantic segmentation in the real domain? The answer is yes, thanks to our novel framework AT/DT. In AT/DT we first learn a mapping function in the synthetic domain between deep features of two networks trained for depth estimation and semantic segmentation. This mapping function captures the relationship between the two tasks. Once learned, we use the mapping on depth features extracted from real samples to solve semantic segmentation in the real domain without the need of labels for it, thereby transferring knowledge across tasks and domains. To further improve performance, we propose two strategies aimed at increasing the transferability of the learned features, namely leveraging on a norm discrepancy alignment loss and an auxiliary task.

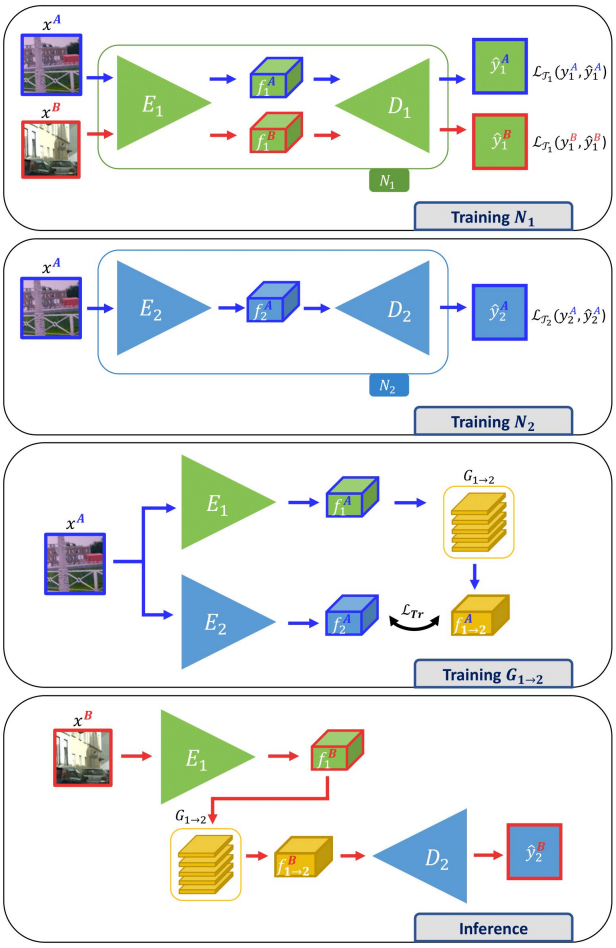


Fig. 2. AT/DT framework: here N_1 and N_2 are trained separately to solve tasks $\mathcal{T}_1$ and $\mathcal{T}_2$. While N_2 is trained only on images from domain $\mathcal{A}$, N_1 is trained jointly on both domain $\mathcal{A}$ and domain $\mathcal{B}$, to enable the extraction of domain invariant features. Then, encoders from the two networks are frozen and used to learn the transfer function $G_{1\to2}$, which aims at transforming features extracted for $\mathcal{T}_1$ in features that are good for $\mathcal{T}_2$. This step is performed only on domain $\mathcal{A}$, since we have no supervision for $\mathcal{T}_2$ on domain $\mathcal{B}$. Finally, at inference time, features are extracted from E_1 starting from images of domain $\mathcal{B}$, transformed with the $G_{1\to2}$ and fed to D_2 to produce the final predictions.

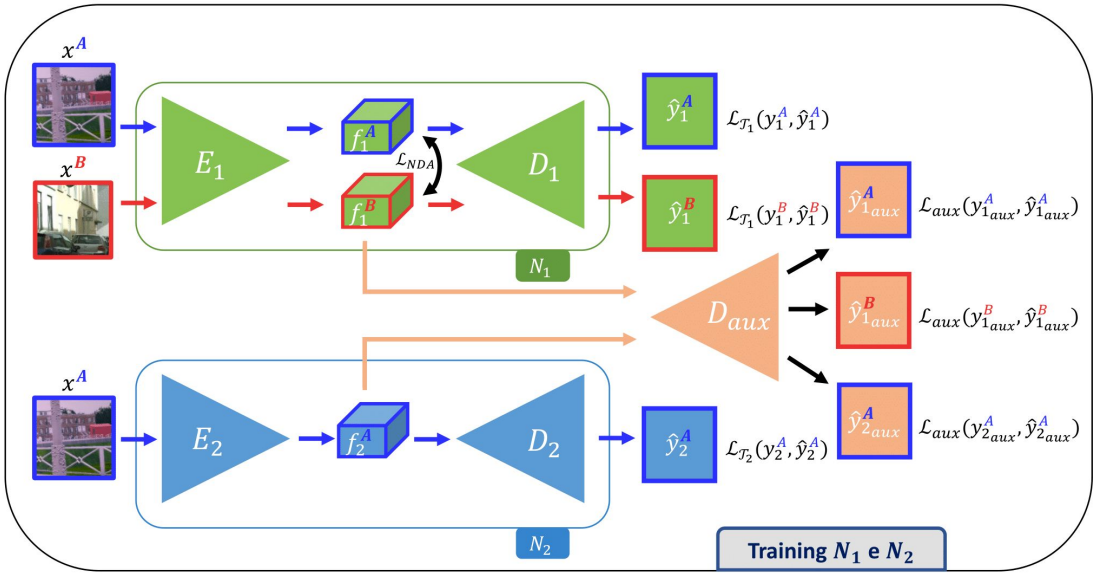


Fig. 3. Features alignment strategies across tasks and domains. We train jointly the networks N_1, N_2 and a shared auxiliary decoder D_{aux}. We train N_1 to solve $\mathcal{T}_1$ on images from domains $\mathcal{A}$ and $\mathcal{B}$ using a supervised loss $\mathcal{L}_{\mathcal{T}_1}$ for $\mathcal{T}_1$ alongside a novel feature Norm Discrepancy Alignment loss $\mathcal{L}_{NDA}$ which helps better aligning the features computed by N_1 across the two domains. We train N_2 using a supervised loss $\mathcal{L}_{\mathcal{T}_2}$ for $\mathcal{T}_2$ on images from $\mathcal{B}$. D_{aux} is trained to solve an auxiliary task $\mathcal{T}_{aux}$ using the loss $\mathcal{L}_{aux}$ and based on the features computed by E_1 on images from $\mathcal{A}$ and $\mathcal{B}$ as well as by E_2 on images from $\mathcal{B}$.

In the following sub-sections, we first describe the base AT/DT framework and then delineate its improved formulation which includes the norm discrepancy alignment loss and auxiliary task.

3.1 Notation

We consider two tasks, $\mathcal{T}_1$ and $\mathcal{T}_2$, as well as two domains, $\mathcal{A}$ and $\mathcal{B}$. We denote the images belonging to $\mathcal{A}$ and $\mathcal{B}$ as $x^{\mathcal{A}}$ and $x^{\mathcal{B}}$, respectively. We have labels for $\mathcal{T}_1$ in $\mathcal{A}$ and $\mathcal{B}$, denoted as $y_1^{\mathcal{A}}$ and $y_1^{\mathcal{B}}$, respectively. On the other hand, we have labels for $\mathcal{T}_2$ only in $\mathcal{A}$, denoted as $y_2^{\mathcal{A}}$. Our aim is to solve $\mathcal{T}_2$ in $\mathcal{B}$, where we do not have supervision. We assume $\mathcal{T}_1$ and $\mathcal{T}_2$ to be both dense tasks, which can therefore be addressed by an encoder-decoder architecture. We denote as N_1 and N_2 two networks that solve $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively. Each network $N_k, k \in \{1, 2\}$ consists of an encoder E_k and a decoder D_k, such that $N_k(x) = D_k(E_k(x))$, x being the input image.

3.2 Across Tasks and Domains Transfer

In our AT/DT framework we aim at learning the relationships between $\mathcal{T}_1$ and $\mathcal{T}_2$ through a neural network. This is achieved by 3 steps, each represented as a block in Figure 2:

Training N_1 and N_2. We train N_1 and N_2 to solve $\mathcal{T}_1$ and $\mathcal{T}_2$. Since we assume supervision for $\mathcal{T}_1$ on both domains, N_1 is trained with images from $\mathcal{A}$ and $\mathcal{B}$. This enables N_1 to learn a feature space shared across the two domains. N_2, instead, is trained only on $\mathcal{A}$. Both networks are trained with a specific supervised task loss $\mathcal{L}_{\mathcal{T}_k}$ for $\mathcal{T}_k$.

Training $G_{1\to2}$. Considering only domain $\mathcal{A}$, where we have supervision for both tasks, we then train a transfer network $G_{1\to2}$ to map the features computed by N_1, $f_1^{\mathcal{A}} = E_1(x^{\mathcal{A}})$, into those computed by N_2, $f_2^{\mathcal{A}} = E_2(x^{\mathcal{A}})$. Denoting the transferred features as $f_{1\to2}^{\mathcal{A}} = G_{1\to2}(f_1^{\mathcal{A}})$, we train the transfer network by minimizing the L_2 loss:

$$\mathcal{L}_{Tr} = ||f_{1\to2}^{\mathcal{A}} - f_2^{\mathcal{A}}||_2 \tag{1}$$

Inference. Once $G_{1\to2}$ has been trained, we can address $\mathcal{T}_2$ in $\mathcal{B}$ by computing the features to solve $\mathcal{T}_1$, $f_1^{\mathcal{B}} = E_1(x^{\mathcal{B}})$, transform them into features amenable to $\mathcal{T}_2$, $f_{1\to2}^{\mathcal{B}} = G_{1\to2}(f_1^{\mathcal{B}})$, and finally decode these features into the required dense output by D_2:

$$\hat{y}_2^B = D_2(f_{1\to2}^{\mathcal{B}}) \tag{2}$$

After presenting the base AT/DT framework, in the next sub-sections we will describe two strategies deployed to boost the feature alignment across domains and tasks. Figure 3 provides a detailed view of these two strategies which in our final proposed framework replace the initial steps of the training protocol (i.e., Training N_1 and N_2).

3.3 Feature Alignment Across Domains

For the effectiveness of the approach delineated in subsection 3.2, it is crucial that $G_{1\to2}$ can generalize well to the target unseen domain $\mathcal{B}$ even if trained only with data from the source domain $\mathcal{A}$.

The DA literature presents us with several ways to accomplish this. One may operate on the input space [27], on the feature space [23] or on the output space of the network [29]. In our setting, though, both input and output space of $G_{1\to2}$ are high dimensional latent spaces and, as reported in [29], unsupervised domain adaptation techniques tend to fail when applied to such spaces while addressing dense tasks. Yet, we can address the domain shift issue with a

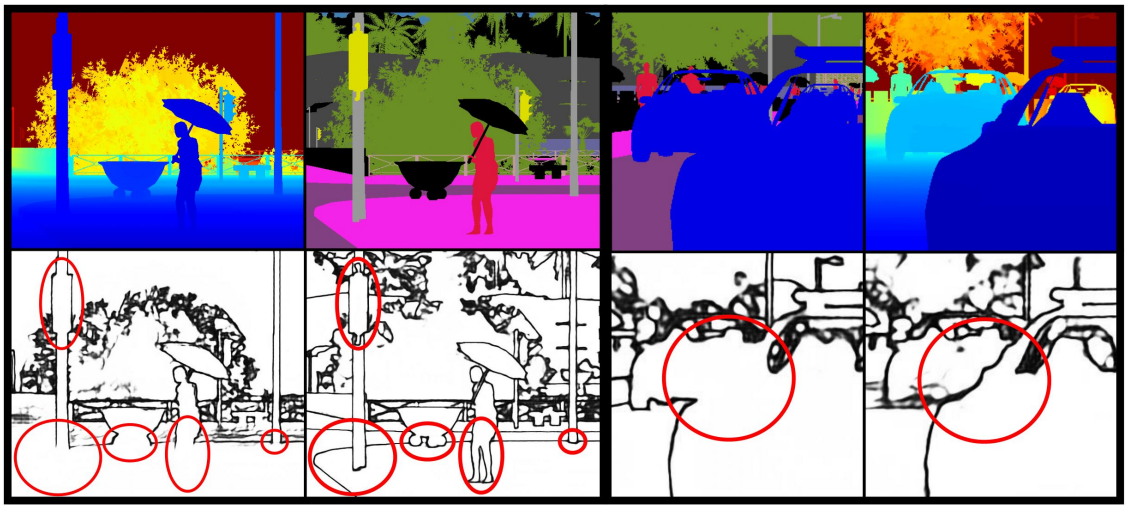


Fig. 4. Two task transfer scenarios: depth-to-semantic on the left, the opposite on the right. First row: ground-truth depth and semantic segmentation maps; second row: corresponding edge maps. Red circles highlight information needed in the target task but missing in the source one.

direct approach in the input space of $G_{1\to2}$, i.e., the feature space of N_1, which is already shared between $\mathcal{A}$ and $\mathcal{B}$ due to the network being trained supervisedly with images from both domains. We leverage on the intuition that scene spatial priors are typically domain invariant in many adaptation scenarios. We consider it as a reasonable assumption for several domain adaptation settings, where we select the source domain by considering visual similarities with the target domain. For instance, in autonomous driving scenarios we typically have cameras placed from a car viewpoint, and scenes are urban scenarios in both synthetic [9], [51] and real [10], [52], [53] datasets. Thus, if we consider the task of semantic segmentation in all datasets (synthetic and real) we typically find *road* pixels in the bottom part of the images and instead sky pixels in the top part of the images. To visualize this property we select a synthetic domain $\mathcal{A}$ CARLA [9] and a real domain $\mathcal{B}$ Cityscapes [10]. Then, we count for each pixel location the number of occurrences of each class. We show the result of this experiment in Figure 5, using a *viridis* colormap to display these occurrency maps for each class and for both domains $\mathcal{A}$ and $\mathcal{B}$. We can clearly see that the maps have a structure similar across domains, e.g., building are concentrated in the top image regions.

Leveraging this property, we propose to align more closely the features computed by E_1 on the images from both domains, i.e., f_1^A and f_1^B, by enforcing similarity of the L_2 norms across channels at the same spatial location. Starting from features f_1^A and f_1^B of dimensionality $H \times W \times C$, where H, W and C are the height, width and number of channels of the feature maps, we calculate the L_2 norm along the C axis and minimize the absolute difference at each spatial location i, j. Hence, our NDA (Norm Discrepancy Alignment) Loss is defined as follows:

$$\mathcal{L}_{NDA} = \frac{1}{W \times H} \sum_{i=1}^{H} \sum_{j=1}^{W} \left| \|f_{1_{i,j}}^A\|_2 - \|f_{1_{i,j}}^B\|_2 \right| \quad (3)$$

3.4 Feature Alignment Across Tasks

While the NDA loss presented above aims at improving the generalization across domains of the feature mapping network $G_{1\to2}$, its effectiveness can be further improved by aligning features also across tasks. Accordingly, we conjecture that f_1 features should capture as much information as possible on the details of the scene, even though some of this information may not be necessary to solve $\mathcal{T}_1$, because, when transferred by $G_{1\to2}$, such a richer representation could help to solve $\mathcal{T}_2$ more effectively. For this reason, while training N_1 for $\mathcal{T}_1$, we train jointly an additional decoder, D_{aux}, to solve an auxiliary task, $\mathcal{T}_{aux}$, aimed at enriching the learnt representation f_1 . However, though multi-task learning of $\mathcal{T}_1$ and $\mathcal{T}_{aux}$ could help to encode more detailed information into f_1 features, it does not guarantee that the decoder D_2, used at inference time on the features $f_{1\to2}$ transferred from $\mathcal{T}_1$ to $\mathcal{T}_2$, may effectively deploy this additional information if it has been trained only to solve $\mathcal{T}_2$ in isolation. This leads us to reckon that D_{aux} should be trained jointly with N_2 too, such that the additional information required to solve $\mathcal{T}_{aux}$ may be incorporated also within the features f_2 learnt by E_2.

Therefore, given auxiliary task labels y_{aux}^A and y_{aux}^B for $\mathcal{A}$ and $\mathcal{B}$, we train N_1 and N_2 jointly with a single auxiliary decoder D_{aux} using an auxiliary loss $\mathcal{L}_{aux}$. Purposely, we obtain auxiliary predictions from both encoders with the shared decoder D_{aux} as $\hat{y}_{k_{aux}} = D_{aux}(E_k(x))$, $k \in \{1, 2\}$. Similarly to the simpler formulation of our framework presented in subsection 3.2, to compute the auxiliary loss we feed images of both domains through E_1, while we pass only images from $\mathcal{A}$ through E_2. We do not pass images belonging to $\mathcal{B}$ through E_2 while training D_{aux} since this would be the only kind of supervision for E_2 in $\mathcal{B}$ and it may skew E_2 output to be more effective on $\mathcal{T}_{aux}$ than on $\mathcal{T}_2$.

3.5 Overall N_1 and N_2 loss

When training simultaneously N_1 and N_2, the overall loss is:

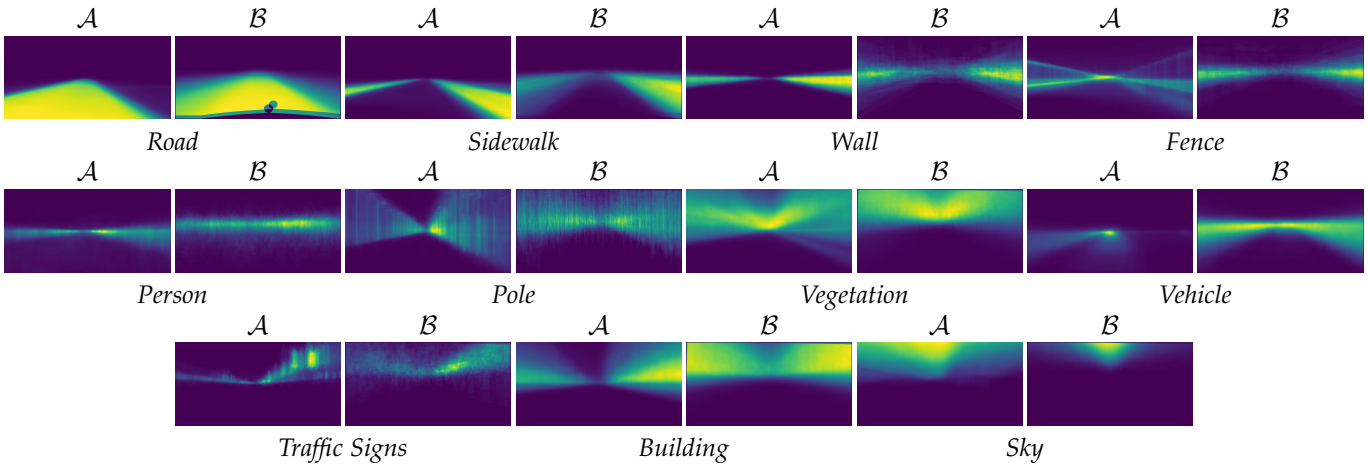


Fig. 5. Spatial Priors Similarities Across Domains. Considered the semantic segmentation task, we compute the number of occurrences of each class at each pixel location for both domains. Domain $\mathcal{A}$ is CARLA, $\mathcal{B}$ is Cityscapes. We visualize the occurrence maps with a *viridis* colormap.

$$\begin{aligned}\mathcal{L} = \lambda_{T_1}\mathcal{L}_{T_1}(y_1^A, \hat{y}_1^A) + \lambda_{T_1}\mathcal{L}_{T_1}(y_1^B, \hat{y}_1^B) \\ +\lambda_{T_2}\mathcal{L}_{T_2}(y_2^A, \hat{y}_2^A) + \lambda_{aux}\mathcal{L}_{aux}(y_{1_{aux}}^A, \hat{y}_{1_{aux}}^A)+ \\ \lambda_{aux}\mathcal{L}_{aux}(y_{1_{aux}}^B, \hat{y}_{1_{aux}}^B) + \lambda_{aux}\mathcal{L}_{aux}(y_{2_{aux}}^A, \hat{y}_{2_{aux}}^A)+ \\ \lambda_{NDA}\mathcal{L}_{NDA}(f_1^A, f_1^B)\end{aligned} \tag{4}$$

4 Experimental Settings

Tasks. We fix $\mathcal{T}_1$ and $\mathcal{T}_2$ to be monocular depth estimation and semantic segmentation, or vice versa. These two visual tasks can be addressed using the same encoder-decoder architecture, with changes needed only in the final layer. Semantic segmentation is solved by minimizing a pixel-wise cross entropy loss, monocular depth estimation by minimizing an L_1 loss. We select edge detection as our $\mathcal{T}_{aux}$ since it seems particularly amenable to improve the effectiveness of our framework in capturing and transferring important structural information that might otherwise be lost. Let us consider the case of $\mathcal{T}_1$ being depth estimation and $\mathcal{T}_2$ semantic segmentation. The features f_1 needed to compute depth may ignore the boundaries between semantically distinct regions showing up at the same distance from the camera: in Figure 4 (left) this is the case, e.g., of the boundaries between legs or tyres and ground, as well as between street signs and poles. Therefore, even if fed to a perfect $G_{1\to 2}$, f_1 may not contain all the information needed to restore the semantic structure of the image. By solving jointly edge detection on the input image, instead, we force our N_1 network to extract additional information that would not need to be captured should the learning objective be concerned with depth estimation only. Similarly, Figure 4 (right) highlights how depth discontinuities do not necessarily correspond to semantic boundaries, such that a network N_1 trained in isolation to assign semantic labels to pixels may not need to learn information relevant to estimate the depth structure of the image. Besides, it is worth pointing out that edge detection can be solved using again the same decoder architecture as $\mathcal{T}_1$ and $\mathcal{T}_2$. Since the edge proxy-labels that we adopt are gray-scale images [6], in our experiments we implement the $\mathcal{L}_{aux}$ loss introduced in subsection 3.4 as a standard L_2 loss. In all our experiments we set λ_{aux} to 0.5, λ_{NDA} to 0.001, λ_{T_1} and λ_{T_2} to 1 to balance loss values.

Datasets. We test the effectiveness of our method in an autonomous driving scenario. We set $\mathcal{A}$ and $\mathcal{B}$ to be a synthetic and a real dataset, respectively. The former consists of a collection of images generated with the Carla simulator [9], while the latter is the popular Cityscapes dataset [10]. We generated the Carla dataset mimicking the camera settings of the real scenes. We render 3500, 500, and 1000 images for training, validation, and testing, respectively. For each image, we store the associated depth and semantic labels provided by the simulator. The Cityscapes dataset is a collection of 2975 and 500 images to be used for training and validation, respectively. As for our evaluation, we use the 500 Cityscapes validation images since test images are not equipped with labels. Moreover, as in Cityscapes only the semantic labels are provided, we use depth proxy-labels obtained with the SGM stereo algorithm [54], by filtering the erroneous predictions in the generated disparities with a left-right consistency check. This can be considered as an added value because it shows the ability to transfer knowledge when learning from noisy labels. Finally, we use a pre-trained[1] state-of-the-art neural network [6] as an off-the-shelf edge detector to extract from the images belonging to $\mathcal{A}$ and $\mathcal{B}$ the edges used as proxy-labels to train $\mathcal{T}_{aux}$.

Architecture. To solve each task, we use two dilated ResNet50 [55] as encoder and a stack of bilinear upsample plus convolutional layers as decoder. The encoder shrinks both input dimensions with a factor of 1/16, while the decoder upsamples the feature map until a prediction with the same spatial resolution as the input image is obtained. The two networks for $\mathcal{T}_1$ and $\mathcal{T}_2$ are identical, but for the final prediction layer, which is task dependent. The two previously defined encoders are also used to capture good features for edge detection, which is solved using D_{aux}, that shares the same architecture as the decoders used in

1. Neither $\mathcal{A}$ nor $\mathcal{B}$ belong to the training set of this network.

N_1 and N_2. $G_{1\to2}$ is a simple CNN made out of 6 pairs of convolutional and batch normalization layers with kernel size 3×3 which do not perform any downsampling or upsampling operation.

Training and Evaluation Protocol. During the training phase of the transfer network $G_{1\to2}$, the model is evaluated on the validation set of Carla. Of course, it is possible that optimality on Carla does not translate into optimal performance on Cityscapes. Yet, we cannot use data from the target domain neither for hyper-parameters tuning nor for early stopping, because in our setting these data would not be available in any real scenario. Therefore, the Cityscapes validation set is only used at test time to measure the final performances of our framework method.

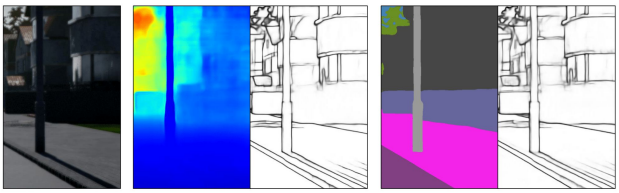


Fig. 6. From left to right: RGB input image of domain $\mathcal{A}$, depth prediction from N_1, edges from f_1, semantic segmentation from N_2 and edges from f_2. Task features f_1 and f_2 encode richer details than strictly needed to solve either tasks as we can recover all edges from both of them by D_{aux}.

Metrics. To evaluate the performance on the semantic segmentation task two metrics are used: pixel accuracy, shortened *Acc.* (i.e the percentage of pixels with a correct predicted label) and Mean Intersection Over Union, shortened *mIoU*, as defined in [10]. To render these metrics comparable among the used datasets, we solve semantic segmentation on the 10 shared classes (Road, Sidewalk, Walls, Fence, Person, Poles, Vegetation, Vehicles, Traffic Signs, Building) plus the Sky category, which is defined as the set of points with infinite depth. Some of the Cityscapes classes are collapsed into one class: car and bicycle into vehicle, traffic signs and traffic light into traffic sign. The remaining categories of Cityscapes are instead ignored.
When testing the depth estimation task, we report the standard metrics described in [57]: Absolute Relative Error (Abs Rel), Square Relative Error (Sq Rel), Root Mean Square Error (RMSE), logarithmic RMSE and δ_1, δ_2 and δ_3 accuracy scores. Each δ_α is obtained by computing, for each pixel of the input image, the maximum among ratio and inverse ratio between the predicted value and the ground-truth. δ_α represents the percentage of pixels whose such ratio is lower than 1.25^α.

5 Experimental Results

We provide results for two different settings: transferring features from depth estimation to semantic segmentation (subsection 5.1) as well as from semantic segmentation to depth estimation (subsection 5.2).

In both scenarios, as already mentioned, we used edge detection as auxiliary task, motivated by the idea that either semantic segmentation and depth estimation can benefit from edge information. Figure 6 shows that with our multi-task learning protocol we are able to restore all the details of the scene from both f_1 and f_2, proving that N_1 and N_2 have indeed learned to encode into their features richer information than that strictly needed to solve $\mathcal{T}_1$ and $\mathcal{T}_2$.

5.1 Depth to Semantics

In this setup, denoted as $Dep \to Sem$, the goal of our framework is to transform depth features into semantic segmentation features. This mapping is learned using Carla as domain $\mathcal{A}$ and Cityscapes as domain $\mathcal{B}$. We report results in Table 1: the first row shows results obtained with no adaptation (i.e., training N_2 on Carla and testing it directly on Cityscapes), while from the second row we can see that our final framework yields 51.28% mIoU and 87.57% Acc with an improvement of +12.48% and +8.99% wrt to the baseline.

Even though AT/DT is the first work to address the across tasks and domains scenario, we compare it against a related work, ZDDA [56], which also leverage auxiliary data from a different tasks to perform domain adaptation. We apply it in our setup using as the "Source" and "Target" domains Carla and Cityscapes respectively. We address the $Dep \to Sem$ scenario using depth maps as "task-irrelevant" data. We skip the last sensor fusion step (Step 3) because it was not applicable in our scenario since we do not have task-irrelevant data at test time, and thus we stop training after the adaptation step (Step 2). We report results of this alternative approach in the second row of Table 1. As we can notice, ZDDA is effective in our scenario and achieves better performance compared to the baseline. However, AT/DT obtains much better results, surpassing ZDDA in all metrics. This is not surprising since ZDDA focus on extracting features only from task-irrelevant data, which can be sub-optimal for the relevant task as these data do not provide the same amount of information as the task-relevant data, e.g., features extracted only from depth images would not contain several useful information for semantic segmentation such as colors or textures.

Furthermore, as we are transferring features from another task, it is worth trying to investigate on the upper bound in performance due to the inherent transferability of the features between the two tasks. Purposely, we train $G_{1\to2}$ using only Cityscapes to learn a mapping function in a supervised fashion as explained in subsection 3.4 on $\mathcal{B}$ and test on the validation set of $\mathcal{B}$. These results are shown in the third row of the table (denoted as Transfer Oracle): given a transfer architecture, there seems to be an upper bound in performance due to the nature of the two tasks, which in the considered setting amounts to a 58.5% mIoU. Thus, our proposal exhibit a gap wrt the Transfer Oracle that is only about -7.2% mIoU. We also report the performance of N_2 trained on $\mathcal{B}$ and tested on $\mathcal{B}$, i.e., the absolute upper bound in performance (last row of the table, denoted as Oracle).

Some qualitative results dealing with the $Dep \to Sem$ scenario are depicted in Figure 7. It is possible to appreciate the overall improvement of our method wrt the baseline, either in flat areas (e.g., roads, sidelwalks and walls), objects shapes (e.g., cars and persons) and fine-grained details (e.g., poles and traffic signs).

TABLE 1
Experimental results of $Dep \rightarrow Sem$ scenario. Baseline stands for N_2 trained on $\mathcal{A}$ and tested on $\mathcal{B}$, Transfer Oracle represents $G_{1\rightarrow 2}$ trained only on $\mathcal{B}$, Oracle refers to N_2 trained and tested on $\mathcal{B}$. Best results highlighted in bold.

$\mathcal{A}$	$\mathcal{B}$	Method	Road	Sidewalk	Walls	Fence	Person	Poles	Vegetation	Vehicles	Tr. Signs	Building	Sky	**mIoU**	**Acc**
Carla	CS	Baseline	78.99	38.81	1.34	5.80	24.02	24.47	71.98	52.23	5.57	65.17	59.10	38.86	78.58
Carla	CS	ZDDA [56]	85.93	41.28	4.62	8.63	38.80	25.94	72.78	58.37	18.44	73.74	**78.16**	46.06	82.82
Carla	CS	**AT/DT**	**90.57**	**48.46**	**7.37**	**12.27**	**41.16**	**31.90**	**81.96**	**72.77**	**23.44**	**77.85**	76.33	**51.28**	**87.57**
CS	CS	Transfer Oracle	89.69	48.05	11.46	29.58	59.68	35.84	85.83	85.57	34.03	78.17	85.54	58.50	88.84
-	CS	Oracle	96.74	78.28	29.26	40.78	72.39	51.28	90.69	91.94	58.92	86.33	89.23	71.44	93.90

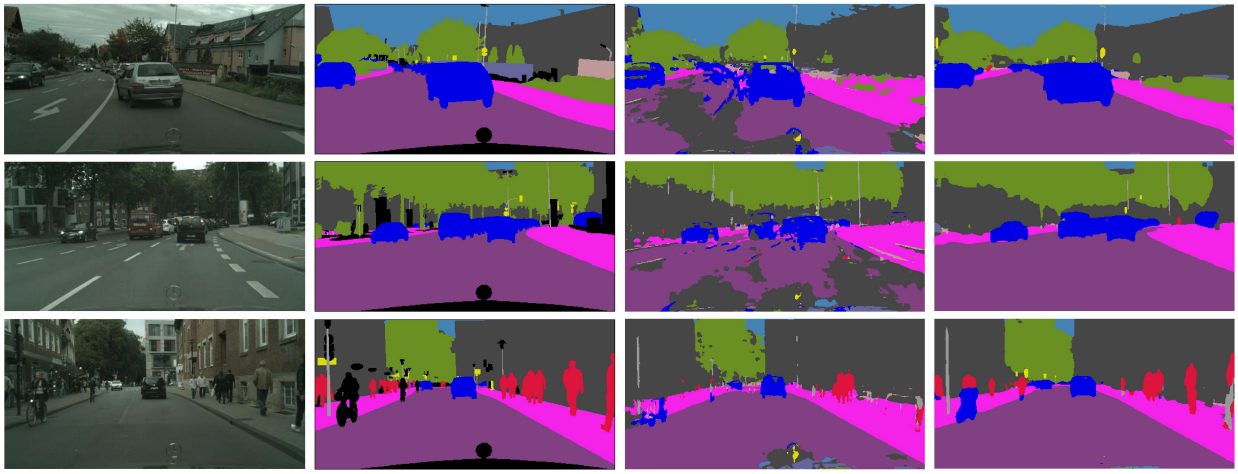


Fig. 7. Qualitative results of the $Dep \rightarrow Sem$ scenario. From left to right: RGB image, ground-truth, baseline trained only on domain $\mathcal{A}$, ours.

5.2 Semantics to Depth

In this setup, which we define as $Sem \rightarrow Dep$, the goal of our framework is to transform semantic features into depth features. This mapping is learned using Carla as domain $\mathcal{A}$ and Cityscapes as domain $\mathcal{B}$, as done for the $Dep \rightarrow Sem$ scenario. Results are reported in Table 2. Similarly to the $Dep \rightarrow Sem$ scenario, in the first row we show results with no adaptation (i.e., our baseline), while the third row presents the ones obtained with our framework. Also for this setup we report performances of ZDDA [56] (second row), in which we use semantic maps as task-irrelevant data. We can see that ZDDA achieves slight better performance of the baseline in 5 metrics out of 7, but still inferior to our approach. Moreover, we report results from the Transfer Oracle and the Oracle, implemented as described for the $Dep \rightarrow Sem$ scenario. It is possible to appreciate that our framework outperforms the baseline on 6 out of 7 metrics, closing remarkably the gap with the practical upper bound of the Transfer Oracle. In Figure 8, we show some qualitative results of the $Sem \rightarrow Dep$ scenario. While predictions look quite noisy in the background, we can see a good improvement in the foreground area thanks to our method. Shapes are recovered almost perfectly, both for big and small objects, even with difficult subjects like the crowd in the bottom row. It is also worth pointing out that the depth predictions yielded by our method turn out much smoother than the ones produced by the baseline and generally less noisy than the ground-truth that, as explained in section 4, consists of proxy-labels computed with SGM [54].

6 Ablation Studies

In the following sections, we study the effectiveness of the key design choices behind our proposal.

6.1 Contribution of $\mathcal{T}_{aux}$ and NDA Loss

We start by studying the effect of introducing in our framework the auxiliary task and the NDA loss, analyzing their contribution when used separately as well as when combined together. The second and third row of Table 3 report the results obtained in the $Dep \rightarrow Sem$ setting by integrating in our method either the auxiliary task (i.e., edge detection) or the NDA loss, respectively. We can see that both design choices bring in an improvement of about +2% in terms of mIoU with respect to the base AT/DT framework (first row). Moreover, the last row of the table shows that the auxiliary edge detection task and the NDA loss turn out complementary because, when combined together, they can provide an overall improvement of +3.34% mIoU.

Figure 9 presents some zoomed-in qualitative results: we can see that, even if the base version of AT/DT already produces satisfactory results at a coarse level, the complete version of our framework can produce much more accurate predictions, especially regarding small details, such as poles, traffic signs and car outlines.

TABLE 2
Experimental results of $Sem \rightarrow Dep$ scenario. Baseline stands for N_2 trained on $\mathcal{A}$ and tested on $\mathcal{B}$, Transfer Oracle represents $G_{1\rightarrow 2}$ trained only on $\mathcal{B}$, Oracle refers to N_2 trained and tested on $\mathcal{B}$. Best results highlighted in bold.

$\mathcal{A}$	$\mathcal{B}$	Method	Lower is better				Higher is better		
			Abs Rel	Sq Rel	RMSE	RMSE log	δ_1	δ_2	δ_3
Carla	CS	Baseline	0.7398	15.169	14.774	0.641	**0.406**	0.650	0.781
Carla	CS	ZDDA [56]	0.5206	7.5491	13.347	0.633	0.345	0.638	0.858
Carla	CS	**AT/DT**	**0.3928**	**4.9094**	**12.363**	**0.444**	0.372	**0.757**	**0.923**
CS	CS	Transfer Oracle	0.2210	2.2962	9.032	0.275	0.669	0.914	0.972
-	CS	Oracle	0.1372	1.6214	8.566	0.244	0.816	0.938	0.976

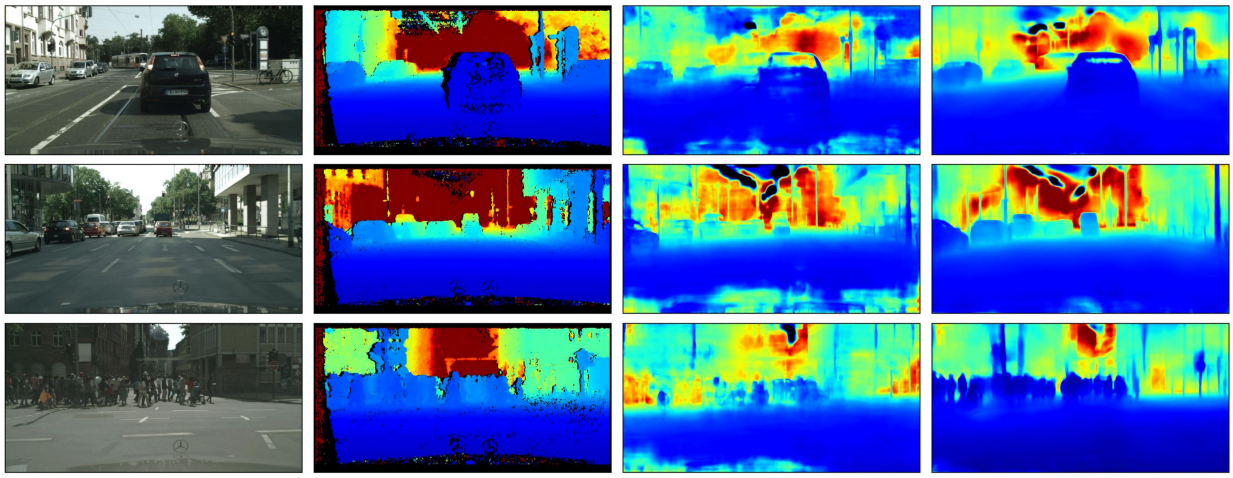


Fig. 8. Qualitative result of the $Sem \rightarrow Dep$ scenario. From left to right: RGB image, ground-truth, baseline network trained only on domain $\mathcal{A}$, ours.

TABLE 3
Ablation study in the $Dep \rightarrow Sem$ scenario. Best results highlighted in bold. Aux refers to the framework trained with the auxiliary task. NDA refers to the framework trained with our NDA loss.

$\mathcal{A}$	$\mathcal{B}$	Aux	NDA	Road	Sidewalk	Walls	Fence	Person	Poles	Vegetation	Vehicles	Tr. Signs	Building	Sky	**mIoU**	**Acc**
Carla	CS			89.95	46.77	5.16	10.21	28.93	28.92	77.50	71.37	19.24	75.29	75.12	48.04	85.90
Carla	CS	✓		90.12	48.90	4.18	11.63	37.40	31.98	**82.34**	71.50	15.11	**78.04**	**80.61**	50.16	87.21
Carla	CS		✓	**91.21**	**50.16**	5.14	**13.78**	36.99	**32.10**	77.72	**73.38**	**23.47**	76.67	72.67	50.30	86.77
Carla	CS	✓	✓	90.57	48.46	**7.37**	12.27	**41.16**	31.90	81.96	72.77	23.44	77.85	76.33	**51.28**	**87.57**

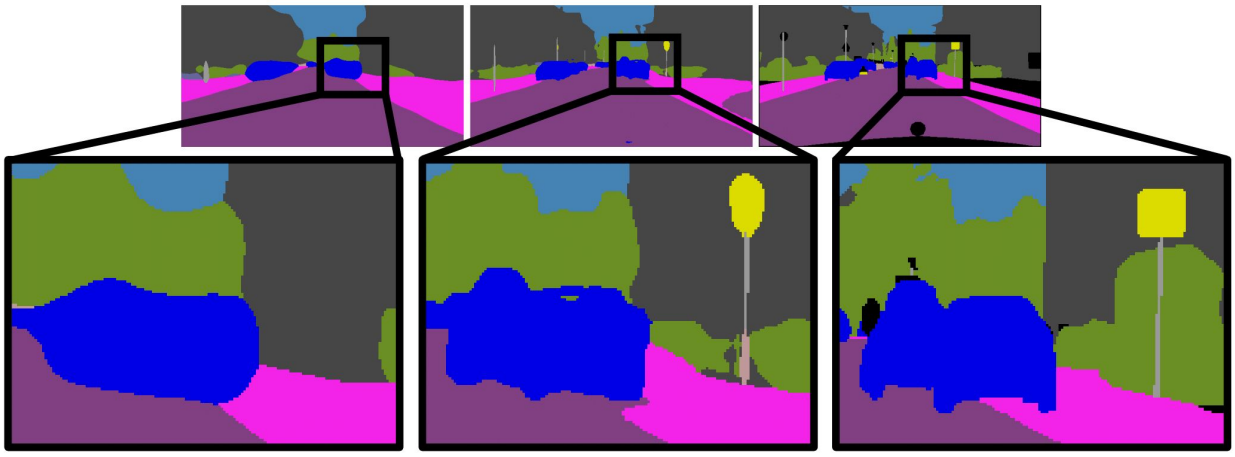


Fig. 9. Zoomed results in a $Dep \rightarrow Sem$ scenario. From left to right: base AT/DT without edge and NDA, our proposed method, ground-truth. We notice how, unlike base AT/DT, our method is able to recover the fine-grained details of the scene.

6.2 Effectiveness of edge detection as auxiliary task

In this section, we show empirically that in our framework the choice of the proper auxiliary task is key to performance.

In both the $Dep \rightarrow Sem$ and the $Sem \rightarrow Dep$ scenarios, we propose to use edge detection as auxiliary task because it captures information about the shapes of the objects in the input images and allows for straightforward computation of proxy-labels. To validate this design choice, we tested our framework in the $Dep \rightarrow Sem$ setting, using D_{aux} to reconstruct the input images both from f_1 and f_2, i.e., the classical autoencoder setting (results in Table 4). Interestingly, using image reconstruction as auxiliary task results in an mIoU score almost identical to the base AT/DT. We consider that the autoencoder task is guided by a reconstruction loss which makes no distinction between the pixels of the input image: such supervision cannot guide effectively f_1 and f_2 to encapsulate the high-frequency components of the image that are needed to predict the fine-grained details of the scene, which is instead obtained by adopting edge detection as auxiliary task.

6.3 Auxiliary tasks as source tasks

The main difference between a source and an auxiliary task is that the auxiliary task alone cannot provide enough information to solve $\mathcal{T}_2$, but it is useful to enrich $\mathcal{T}_1$ features and align feature content across tasks. To better support our claims, we investigated AT/DT behaviour when using auxiliary tasks $\mathcal{T}_{aux}$ as source tasks $\mathcal{T}_1$ and semantic segmentation as target task $\mathcal{T}_2$. The results of these experiments are reported in Table 5. All rows of the table show results of the *base* AT/DT i.e., trained without L_{aux} and L_{NDA} losses. As we can notice, using as source task $\mathcal{T}_1$ a standard image-reconstruction (row 1, autoencoder) or an edge detection (row 2) lead to much worse results than using depth estimation (row 3). We argue that features extracted by N_1 for these tasks do not contain enough information to perform semantic segmentation, which are yet contained in features for depth estimation. Similar finding were also made by Taskonomy [1], in which they show that edge detection and image reconstruction (aka autoencoder) are less correlated to semantic segmentation than depth estimation. On the contrary, we have shown that Edge Detection can be a good auxiliary task in the $Dep \rightarrow Sem$ scenario since it can enrich depth features with missing edges useful for semantic segmentation and it can increase transferability aligning depth and semantic features.

6.4 Importance of simultaneous training of N_1, N_2 and D_{aux}

In our experiments we use edge detection as auxiliary task and train a shared decoder D_{aux} to reconstruct the edges of the input image from the features extracted by both E_1 and E_2. In fact, we argue that this procedure should force E_1 to encode into the extracted features also edge information that may be not necessary to solve $\mathcal{T}_1$ but that may be relevant for $\mathcal{T}_2$. Besides, we believe that simultaneous training of N_1, N_2 and D_{aux} is crucial to encourage features coming from E_1 and E_2 to represent edge information in a similar manner, making it easier to learn $G_{1\rightarrow 2}$.

In Table 6 we report the results concerning the ablation study conducted to validate these intuitions. We consider the $Dep \rightarrow Sem$ scenario using the Carla dataset as domain $\mathcal{A}$ and Cityscapes as domain $\mathcal{B}$. The four rows of the table deal with the following training schemes:

1) The *base* AT/DT (i.e., without $\mathcal{T}_{aux}$ and NDA loss) as baseline.
2) We first train N_1 and D_{aux} on both $\mathcal{A}$ and $\mathcal{B}$. Then, we train N_2 on $\mathcal{A}$. Finally, we train $G_{1\rightarrow 2}$ on features extracted by E_1 and E_2 on domain $\mathcal{A}$.
3) We train N_1 and a first D^1_{aux} on both $\mathcal{A}$ and $\mathcal{B}$. Then, we train N_2 and a second D^2_{aux} on $\mathcal{A}$. Finally, we train $G_{1\rightarrow 2}$ on features extracted by E_1 and E_2 on domain $\mathcal{A}$
4) Our proposed method, which trains N_1, N_2 and a shared D_{aux} simultaneously.

The introduction of edge detection as auxiliary task helps in every scenario. In fact, if we use D_{aux} only while training N_1 (second row), we already see an increase of 0.6% in the overall mIoU. We believe that this is explained by the presence of edge details (not strictly necessary to solve $\mathcal{T}_1$ but relevant for $\mathcal{T}_2$) in the features extracted by E_1. However, $G_{1\rightarrow 2}$ may experience difficulties in adapting f_1 into f_2 if edge information is not explicitly present in f_2. This is confirmed by the results in the third row of the table, where an additional increase of 1.3% in the overall mIoU is attained by using two different D_{aux} (one during training of N_1 and one during training of N_2). Finally, the best results in terms of mIoU and Acc are achieved by our method, i.e., when training N_1, N_2 and a shared D_{aux} simultaneously. This vouches for the benefit of encoding in a similar manner the edge information in f_1 and f_2 in order to enforce feature alignment across tasks.

6.5 Alignment strategies for N_1

An alternative approach to align N_1 features between domains to ease the transfer process and favor the generalization of $G_{1\rightarrow 2}$ consists in leveraging on the widely adopted adversarial training in feature space. In our setting, this can be obtained by adding a critic that must discriminate whether the features produced by E_1 come from $\mathcal{A}$ or $\mathcal{B}$. Thus, the encoder E_1 not only has to learn a good feature space for its task, but it is also asked to fool the critic. Afterwards, we can proceed to learn a mapping function $G_{1\rightarrow 2}$ among tasks as usual. In Table 7 we compare this standard DA methodology to our NDA loss. Adversarial training (second row) does not introduce significant improvements with respect to not performing DA for $\mathcal{T}_1$ (i.e., base AT/DT, first row), while constraining the features extracted by E_1 in a norm aligned space (third row) significantly increases both performance metrics with respect to the baseline. Our intuition is that, although adversarial training can be useful for domain alignment, it alters the learned feature space with the goal of fooling the critic and this training objective can lead to worse performances on the current task. Our NDA loss, on the other hand, acts as a regularizer that favors the learning of an homogeneous latent space across the domains involved in our experiments, improving the generalization capability of the transfer network without

TABLE 4
Comparison between autoencoder and edge detection as auxiliary tasks in the $Dep \rightarrow Sem$ scenario. Best results highlighted in bold.

$\mathcal{T}_{aux}$	Road	Sidewalk	Walls	Fence	Person	Poles	Vegetation	Vehicles	Tr. Signs	Building	Sky	**mIoU**	**Acc**
None	89.95	46.77	5.16	10.21	28.93	28.92	77.50	71.37	**19.24**	75.29	75.12	48.04	85.90
Autoencoder	**90.68**	**50.12**	**7.45**	9.08	31.40	29.43	78.72	68.51	12.95	74.67	75.68	48.07	86.31
Edge detection	90.12	48.90	4.18	**11.63**	**37.40**	**31.98**	**82.34**	**71.50**	15.11	**78.04**	**80.61**	**50.16**	**87.21**

TABLE 5
Auxiliary tasks as source tasks in the $Dep \rightarrow Sem$ scenario. Best results highlighted in bold.

$\mathcal{T}_{aux}$ as $\mathcal{T}_1$	Road	Sidewalk	Walls	Fence	Person	Poles	Vegetation	Vehicles	Tr. Signs	Building	Sky	**mIoU**	**Acc**
Autoencoder	60.24	19.33	1.67	1.67	4.12	8.00	33.15	10.49	0.69	17.89	62.66	19.99	52.91
Edge Detection	63.82	16.60	0.67	1.37	6.55	10.26	47.62	4.42	0.11	33.90	38.87	20.38	58.33
Depth	**89.95**	**46.77**	**5.16**	**10.21**	**28.93**	**28.92**	**77.50**	**71.37**	**19.24**	**75.29**	**75.12**	**48.04**	**85.90**

TABLE 6
Ablation study on the importance of simultaneous training of the $\mathcal{T}_1$, $\mathcal{T}_2$, and the auxiliary task. Best results highlighted in bold. See text for a detailed explanation of the training protocol used in each row.

method	Road	Sidewalk	Walls	Fence	Person	Poles	Vegetation	Vehicles	Tr. Signs	Building	Sky	**mIoU**	**Acc**
base AT/DT	89.95	46.77	5.16	10.21	28.93	28.92	77.50	71.37	**19.24**	75.29	75.12	48.04	85.90
Separate (N_1 + edge), N_2	87.24	43.30	3.08	10.17	41.77	29.04	81.81	72.35	16.58	77.10	73.10	48.69	85.89
Separate (N_1 + edge), (N_2 + edge)	88.83	47.31	**7.10**	8.59	**44.53**	30.99	**83.24**	**73.54**	18.05	**78.10**	69.66	49.99	86.72
Simultaneous (N_1 + N_2 + edge)	**90.12**	**48.90**	4.18	**11.63**	37.40	**31.98**	82.34	71.50	15.11	78.04	**80.61**	**50.16**	**87.21**

degrading the performances in the single tasks. Then, from the third to the fifth row, we compare our NDA loss with another strategy, LargerNorm [24], that also align features across domains operating on the feature norms. They show that features are more transferable across domains if we constrain feature norms to be equal to an arbitrary large number. We notice that the method is very sensible to the norm value, and it could be hard to select without using target labels. When using an appropriate norm value (25, fourth row), the method achieves a slight improvement over the baseline without alignment. However, since it just force all features globally to be a large number, it is not well-suited for tasks in which we have a spatial dimensions such as semantic segmentation. Moreover, in the sixth row, we experiment also with a more recent adversarial loss formulation, Asymmetric Adv. [58], which preserve discriminability while performing domain alignment by changing only target features instead of both source and target ones. However, we notice that this method is achieving the worst results among feature alignment strategies, even worse than the baseline. Our motivation is that aligning feature distribution in such a high dimensional feature space with a spatial structure might be too difficult to achieve by only changing target features. Finally, we notice that NDA achieves the best performance probably because it only align features norm rather than the whole marginal distribution, which is an easier goal that can be achieved also in high-dimensional space. Moreover, NDA operates at each spatial location independently rather than globally, exploiting the spatial priors similarity across domains, reaching better performances.

6.6 Aligning N_2 features

We tried to perform feature alignment across domains also on the features f_2 extracted by E_2, either by deploying adversarial training or imposing our NDA loss. The idea is to favor the generalization of $G_{1\rightarrow 2}$ by making more homogeneous not only its input space (i.e., the features produced by E_1, aligned with our NDA loss), but also its output space, i.e., the features produced by E_2. However, the setting is not completely symmetric: when learning E_2, we do not have supervision available for $\mathcal{B}$, and the only loss shaping the feature space for its images would be the alignment loss. We report results of this ablation study in Table 8 and discuss them below.

In the first row, we report the results provided by the *base* AT/DT (without L_{NDA} and L_{aux}). In the following two rows, we show results obtained by an adversarial (row 2) and an asymmetric adversarial [58] (row 3) training

TABLE 7
Comparison between NDA loss and other strategies to align E_1 features. Best results highlighted in bold.

E_1 Align.	Road	Sidewalk	Walls	Fence	Person	Poles	Vegetation	Vehicles	Tr. Signs	Building	Sky	**mIoU**	**Acc**
None	89.95	46.77	5.16	10.21	28.93	28.92	77.50	71.37	19.24	75.29	75.12	48.04	85.90
Adv.	89.89	46.01	4.22	11.89	**38.20**	30.65	77.00	63.68	12.99	74.35	**81.16**	48.19	85.42
LargerNorm [24] (1)	38.37	24.17	0.56	3.66	10.50	23.04	52.61	9.41	3.42	52.64	10.54	20.81	51.49
LargerNorm [24] (25)	86.82	42.23	1.94	9.00	34.92	29.02	76.39	70.97	23.38	74.97	80.00	48.15	84.62
LargerNorm [24] (500)	78.94	31.25	2.53	6.00	22.08	20.55	68.18	26.21	4.35	62.28	63.53	35.08	76.53
Asymmetric Adv. [58]	86.69	38.57	**5.92**	5.72	27.43	22.91	70.81	70.71	7.86	72.15	75.18	44.00	83.38
NDA	**91.21**	**50.16**	5.14	**13.78**	36.99	**32.10**	**77.72**	**73.38**	**23.47**	**76.67**	72.67	**50.30**	**86.77**

TABLE 8
Results of aligning output space of E_2 in a $Dep \rightarrow Sem$ scenario. Best results highlighted in bold.

E_2 Align.	Road	Sidewalk	Walls	Fence	Person	Poles	Vegetation	Vehicles	Tr. Signs	Building	Sky	**mIoU**	**Acc**
None	**89.95**	**46.77**	5.16	**10.21**	28.93	28.92	**77.50**	71.37	19.24	**75.29**	75.12	**48.04**	**85.90**
Adv.	89.36	46.03	5.59	8.22	**36.45**	25.44	75.15	**72.29**	12.69	74.12	**75.79**	47.38	85.31
Asymmetric Adversarial [58]	87.90	42.81	**7.64**	8.44	26.02	**29.11**	72.54	69.01	**24.01**	71.71	70.42	46.33	83.61
NDA	44.94	23.82	3.81	2.09	30.74	24.21	42.08	68.84	11.69	35.67	11.10	27.18	56.17

TABLE 9
Results of aligning output space of D_2 in a $Dep \rightarrow Sem$ scenario. Best results highlighted in bold.

D_2 Align.	Road	Sidewalk	Walls	Fence	Person	Poles	Vegetation	Vehicles	Tr. Signs	Building	Sky	**mIoU**	**Acc**
None	**89.95**	**46.77**	**5.16**	**10.21**	**28.93**	**28.92**	**77.50**	**71.37**	**19.24**	**75.29**	**75.12**	**48.04**	**85.90**
Adv.	87.48	45.73	0.63	2.12	26.22	26.39	61.40	66.92	12.97	66.39	74.77	42.82	81.87

TABLE 10
Results of aligning input and/or output space of $G_{1\rightarrow 2}$ in a $Dep \rightarrow Sem$ scenario. Best results highlighted in bold.

Input Align.	Output Align.	Road	Sidewalk	Walls	Fence	Person	Poles	Vegetation	Vehicles	Tr. Signs	Building	Sky	**mIoU**	**Acc**
-	NDA	42.97	19.60	2.31	1.36	4.21	15.74	18.42	11.77	7.19	36.72	38.99	18.12	43.63
-	Adv	90.80	48.91	**6.16**	11.84	35.32	30.29	**78.78**	71.17	18.51	75.66	75.03	49.32	86.43
	Asymmetric Adv. [58]	85.49	40.70	4.94	10.49	34.02	30.26	76.31	70.30	17.07	74.30	72.94	46.99	83.86
-	NDA + Adv	91.03	48.93	6.14	12.24	35.91	31.05	77.93	70.28	16.65	75.50	74.47	49.10	86.28
-	Adv D2	90.20	47.54	5.92	11.76	**37.03**	29.52	77.98	72.42	19.28	75.82	**77.03**	49.50	86.28
NDA	Adv	90.67	49.49	5.54	12.29	36.73	28.49	78.28	70.19	22.05	76.47	76.35	49.69	86.73
NDA	-	**91.21**	**50.16**	5.14	**13.78**	36.99	**32.10**	77.72	**73.38**	**23.47**	**76.67**	72.67	**50.30**	**86.77**

on the features f_2, using the same procedures described in the previous sub-section for f_1. We can observe that, not only both adversarial trainings does not improve (like adversarial training applied to E_1), but they even decrease the overall mIoU compared to the baseline. Finally, in the fourth row, we report the results obtained by our NDA loss on f_2: the NDA loss destroys the feature space of $\mathcal{T}_2$ when applied in this context, as vouched by the drop of 20% in the overall mIoU wrt to base AT/DT.

During AT/DT inference, we use also D_2 to yield the final task predictions. Nevertheless, D_2 has been trained only on $\mathcal{A}$, thus its performance may be harmed when using $\mathcal{B}$ images. Thus, we ran an additional test reported in Table 9. Following [29] we train N_2 (i.e., E_2 and D_2) using an adversarial loss on the D_2 output space, thus making D_2 aware of $\mathcal{B}$. Then, we train $G_{1\rightarrow 2}$ to map features of E_1 into features of E_2, and during inference we employ the previously trained decoder D_2 to produce the final outputs reporting the results in row *Adv.*. We notice a clear drop in performance w.r.t. *base* AT/DT (row *None*), i.e. AT/DT trained without L_{NDA} and L_{aux}.

We formulate the following hypothesis to explain the

above results: all adversarial trainings and NDA loss try to align $f_2^{\mathcal{A}}$ and $f_2^{\mathcal{B}}$. While $f_2^{\mathcal{A}}$ are shaped also by the supervision of $\mathcal{T}_2$, $f_2^{\mathcal{B}}$ evolve only according to the additional loss we impose, as we do not have supervision for $\mathcal{T}_2$ on $\mathcal{B}$. However, E_2 is shared across domains, and therefore may be pushed to produce worse representations for both domains while it tries to accomplish the adversarial objectives or the NDA loss minimization for $\mathcal{B}$. If this happens, mappings learned by $G_{1\to2}$ from $f_1^{\mathcal{A}}$ to $f_2^{\mathcal{A}}$ will hallucinate worse features for $\mathcal{T}_2$ on $\mathcal{B}$. To understand why adversarial trainings leads to small decreases in performances compared to the use of NDA loss, we ought to consider that adversarial training implies a discriminator that cannot be easily fooled by totally degenerated features, while, without any additional constrain from task supervision, the NDA loss may yield totally collapsed representation.

6.7 Aligning $G_{1\to2}$ features

Although feature alignment does not turn out beneficial when training N_2, one may still expect to obtain better hallucinated features if the representations obtained when transferring $f_1^{\mathcal{A}}$ and $f_1^{\mathcal{B}}$ are aligned. We empirically found out that even though output space aligning strategies deployed when training $G_{1\to2}$ can lead to improvements in performance, input space alignment using our NDA loss deployed when training N_1 is more effective. Moreover, combining input and output space alignment techniques does not lead to further improvements. We performed this ablation study in the $Dep \to Sem$ scenario using Carla as $\mathcal{A}$ and Cityscapes as $\mathcal{B}$. The results of these experiments are reported in Table 10.

First, we applied our NDA loss to the output-space of $G_{1\to2}$. Similarly to what discussed in the previous section, we notice that, without supervision on $\mathcal{B}$, the representations transformed from $G_{1\to2}$ while minimizing the NDA loss yield a drastic drop in the framework performance (row 1). We also tried to align the output space features by training $G_{1\to2}$ alongside a discriminator in an adversarial fashion. We wanted to fool the discriminator in order to generate indistinguishable features from $\mathcal{A}$ or $\mathcal{B}$. We notice that this strategy allows us to reach good overall performances with a 49.32 mIoU on Cityscapes (second row). Moreover, we thought that, as adversarial training provides a supervision on $\mathcal{B}$, using the NDA loss in combination with the adversarial loss could avoid producing degenerated features for $\mathcal{B}$ while reaching a better overall alignment between $\mathcal{A}$ and $\mathcal{B}$. However, we notice that the combination of the two losses leads us to slightly worse results than adversarial training alone (rows 2 vs 3). Furthermore, since using an adversarial loss on the output space of $G_{1\to2}$ lead us to good overall performances, we tested it in combination with the best input space alignment from Table 7, i.e. NDA loss applied when training N_1. However, the combination of these two methods achieves worse performance than using only the NDA loss on input space (rows 6 vs 7). Finally, we also experimented a different alignment strategy for the $G_{1\to2}$ output space. Instead of directly applying adversarial loss in E_2 feature space, we apply adversarial loss in D_2 output space while training $G_{1\to2}$. As discussed in [29], output space is easier to align than feature space for several reasons: i) the scene semantic structure is typically similar across domains ii) the feature space encode many information such as color, light, textures iii) the feature space has higher dimensions. By aligning D_2 output space we indirectly influence also E_2 features making them more domain aligned. During training, we keep D_2 frozen and we update only $G_{1\to2}$ weights. Also in this case, if compare this methodology with simply using L_{NDA} alone (row 6 vs row 7), it achieves worse results.

7 Concluding Remarks

We have introduced a framework to transfer knowledge between different tasks by learning an explicit mapping function between deep features. This mapping function can be parametrized by a neural network and show interesting generalization capabilities across domains. To further ameliorate performance we have proposed two novel feature alignment strategies. At a domain level, we showed that the transfer function presented in our framework can be boosted by making its input space more homogeneous across domains with our simple yet effective NDA loss. At a task level, instead, we reported how deep features extracted for different tasks can be enriched and aligned with the introduction of a shared auxiliary task, which we implemented as edge detection in our experiments. We reported good results in the challenging synthetic to real scenario while transferring knowledge between the semantic segmentation and monocular depth estimation tasks.

Our proposal is complementary to the whole domain adaptation literature and might be integrated with it. While DA directly applied to the learned feature space does not seems effective (see Table 8) more modern techniques either try to align the prediction in the final label space [29] or rely on self-ensembling for pseudo labeling [59]. We plan to incorporate these promising direction into our framework as part of future developments.

References

[1] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3712–3722.

[2] A. R. Zamir, A. Sax, N. Cheerla, R. Suri, Z. Cao, J. Malik, and L. J. Guibas, "Robust learning through cross-task consistency," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 197–11 206.

[3] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, p. 135–153, Oct 2018. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2018.05.083

[4] P. Z. Ramirez, A. Tonioni, S. Salti, and L. D. Stefano, "Learning across tasks and domains," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8110–8119.

[5] P. Z. Ramirez, A. Tonioni, and L. Di Stefano, "Exploiting semantics in adversarial training for image-level domain adaptation," in *2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS)*. IEEE, 2018, pp. 49–54.

[6] X. Soria, E. Riba, and A. Sappa, "Dense extreme inception network: Towards a robust cnn model for edge detection," in *The IEEE Winter Conference on Applications of Computer Vision (WACV '20)*, 2020.

[7] S. Xie and Z. Tu, "Holistically-nested edge detection," *CoRR*, vol. abs/1504.06375, 2015. [Online]. Available: http://arxiv.org/abs/1504.06375

[8] Y. Wang, X. Zhao, Y. Li, and K. Huang, "Deep crisp boundaries: From boundaries to higher-level tasks," *IEEE Transactions on Image Processing*, vol. 28, no. 3, p. 1285–1298, Mar 2019. [Online]. Available: http://dx.doi.org/10.1109/TIP.2018.2874279

[9] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[11] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *arXiv preprint arXiv:1911.02685*, 2019.

[12] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, p. 1137–1149, Jun 2017. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2016.2577031

[16] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2017.322

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[18] A. Pal and V. N. Balasubramanian, "Zero-shot task transfer," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[19] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2066–2073.

[20] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 37. PMLR, 2015, pp. 97–105.

[21] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*, 2015, pp. 1180–1189.

[22] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[23] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[24] R. Xu, G. Li, J. Yang, and L. Lin, "Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1426–1435.

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[26] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[27] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2017.18

[28] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[29] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2018.00780

[30] W. Hong, Z. Wang, M. Yang, and J. Yuan, "Conditional generative adversarial network for structured domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1335–1344.

[31] F. Pizzati, R. d. Charette, M. Zaccaria, and P. Cerri, "Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 2990–2998.

[32] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "Fcns in the wild: Pixel-level adversarial and constraint-based adaptation," *arXiv preprint arXiv:1612.02649*, 2016.

[33] Y. Zhang, P. David, and B. Gong, "Curriculum domain adaptation for semantic segmentation of urban scenes," in *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[34] W.-L. Chang, H.-P. Wang, W.-H. Peng, and W.-C. Chiu, "All about structure: Adapting structural information across domains for boosting semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1900–1909.

[35] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, 10–15 Jul 2018, pp. 1989–1998.

[36] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[37] Y. Zhang, Z. Qiu, T. Yao, D. Liu, and T. Mei, "Fully convolutional adaptation networks for semantic segmentation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2018.00712

[38] S. Sankaranarayanan, Y. Balaji, A. Jain, S. N. Lim, and R. Chellappa, "Learning from synthetic data: Addressing domain shift for semantic segmentation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2018.00395

[39] F. Pan, I. Shin, F. Rameau, S. Lee, and I. S. Kweon, "Unsupervised intra-domain adaptation for semantic segmentation through self-supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3764–3773.

[40] M. Kim and H. Byun, "Learning texture invariant representation for domain adaptation of semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 975–12 984.

[41] A. Tonioni, M. Poggi, S. Mattoccia, and L. Di Stefano, "Unsupervised adaptation for deep stereo," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[42] C. Zheng, T.-J. Cham, and J. Cai, "T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[43] A. Tonioni, F. Tosi, M. Poggi, S. Mattoccia, and L. D. Stefano, "Real-time self-adaptive deep stereo," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[44] K.-H. Lee, G. Ros, J. Li, and A. Gaidon, "SPIGAN: Privileged adversarial learning from simulation," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=rkxoNnC5FQ

[45] I. Kokkinos, "Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2017.579

[46] P. Z. Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. Di Stefano, "Geometry meets semantics for semi-supervised monocular depth estimation," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 298–313.

[47] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,"

2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2018.00781

[48] F. Tosi, F. Aleotti, P. Z. Ramirez, M. Poggi, S. Salti, L. Di Stefano, and S. Mattoccia, "Distilled semantics for comprehensive scene understanding from videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[49] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4068–4076.

[50] J. N. Kundu, N. Lakkakula, and R. V. Babu, "Um-adapt: Unsupervised multi-task adaptation using adversarial cross-task distillation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1436–1445.

[51] S. R. Richter, Z. Hayder, and V. Koltun, "Playing for benchmarks," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 2232–2241. [Online]. Available: https://doi.org/10.1109/ICCV.2017.243

[52] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[53] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[54] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 807–814.

[55] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[56] K.-C. Peng, Z. Wu, and J. Ernst, "Zero-shot deep domain adaptation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 764–781.

[57] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014, pp. 2366–2374.

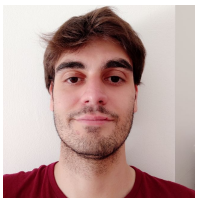
[58] J. Yang, H. Zou, Y. Zhou, Z. Zeng, and L. Xie, "Mind the discriminability: Asymmetric adversarial domain adaptation," in *European Conference on Computer Vision*. Springer, 2020, pp. 589–606.

[59] J. Choi, T. Kim, and C. Kim, "Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 6830–6840.

Pierluigi Zama Ramirez is a Post Doc at the Computer Vision Laboratory (CVLab), University of Bologna. His research interests include deep learning, semantic segmentation, depth estimation, optical flow and domain adaptation. He has authored more than 10 papers on these subjects.

Adriano Cardace is a PhD student at the Computer Vision Laboratory (CVLab), University of Bologna. His research interests include deep learning for Computer Vision problems, especially semantic segmentation, domain adaptation and self-supervised learning.

Luca De Luigi is a PhD student at the Computer Vision Laboratory (CVLab) at the University of Bologna. His research focuses on deep learning for computer vision problems, especially dealing with 3D geometry and implicit neural representations.



Alessio Tonioni received his PhD degree in Computer Science and Engineering from University of Bologna in 2019. Currently, he is a research scientist at Google Zurich. His research interest concerns machine learning for depth estimation, domain adaptation and generalization. He has authored more than 15 papers on these subjects.

Samuele Salti is currently assistant professor at the Department of Computer Science and Engineering (DISI) of the University of Bologna, Italy. Before joining the University of Bologna, he was leading the Data Science team at Verizon Connect, the world leading company in fleet management products and connected vehicles services. His main research interest is computer vision, in particular 3D computer vision, and machine/deep learning applied to computer vision problems. Dr. Salti has co-authored 42 publications in international conferences and journals and 8 international patents. In 2020, he co-founded the start-up eyecan.ai. He was awarded the best paper award runner-up at 3DIMPVT 2011, the top international conference on 3D computer vision, and was nominated outstanding reviewer at CVPR 2020 and NeurIPS 2020.

Luigi Di Stefano received the PhD degree in electronic engineering and computer science from the University of Bologna, in 1994. He is currently a full professor with the Department of Computer Science and Engineering, University of Bologna, where he founded and leads the Computer Vision Laboratory (CVLab). His research interests include image processing, computer vision and machine/deep learning. He is the author of more than 150 papers and several patents. He has been scientific consultant for major companies in the fields of computer vision and machine learning. He is a member of the IEEE Computer Society, IEEE, and the IAPR-IC.