MULTI-TASK LEARNING WITH 3D-AWARE REGULARIZATION

Wei-Hong Li[1], **Steven McDonagh**[1], **Ales Leonardis**[2], **Hakan Bilen**[1]
[1]University of Edinburgh, [2]University of Birmingham
github.com/VICO-UoE/3DAwareMTL

ABSTRACT

Deep neural networks have become the standard solution for designing models that can perform multiple dense computer vision tasks such as depth estimation and semantic segmentation thanks to their ability to capture complex correlations in high dimensional feature space across tasks. However, the cross-task correlations that are learned in the unstructured feature space can be extremely noisy and susceptible to overfitting, consequently hurting performance. We propose to address this problem by introducing a structured 3D-aware regularizer which interfaces multiple tasks through the projection of features extracted from an image encoder to a shared 3D feature space and decodes them into their task output space through differentiable rendering. We show that the proposed method is architecture agnostic and can be plugged into various prior multi-task backbones to improve their performance; as we evidence using standard benchmarks NYUv2 and PASCAL-Context.

1 INTRODUCTION

Learning models that can perform multiple tasks coherently while efficiently sharing computation across tasks is the central focus of multi-task learning (MTL) (Caruana, 1997). Deep neural networks (DNNs), which have become the standard solution for various computer vision problems, provide at least two key advantages for MTL. First, they allow for sharing a significant portion of features and computation across multiple tasks, hence they are computationally efficient for MTL. Second, thanks to their hierarchical structure and high-dimensional representations, they can capture complex cross-task correlations at several abstraction levels (or layers).

Yet designing multi-task DNNs that perform well in all tasks is extremely challenging. This often requires careful engineering of mechanisms that allow for the sharing of relevant features between tasks, while also maintaining task-specific features. Many recent multi-task DNNs (Vandenhende et al., 2021) can be decomposed into shared feature encoder across all tasks and following task-specific decoders to generate predictions. The technical challenge here is to strike a balance between the portion of the shared and task-specific features to achieve good performance-computation trade-off. To enable more flexible feature sharing and task-specific adaptation, Liu et al. (2019) propose to use 'soft' task-specific attention modules appended to the shared encoder that effectively shares most features and parameters across the tasks while adapting them to each task through light-weight attention modules. However, these attention modules are limited to share features across tasks only within each layer (or scale). Recent works (Vandenhende et al., 2020b; Bruggemann et al., 2021) propose to aggregate features from different layers and to capture cross-task relations from the multi-scale features. More recently, capturing long-range spatial correlations across multiple tasks has been shown to further improve MTL performance through use of vision transformer modules (Ye & Xu, 2022).

Our central hypothesis is that the high-dimensional and unstructured features, shared across tasks, in the recent MTL models are prone to capturing noisy cross-task correlations and hence hurt performance. To this end, we propose regulating the feature space of shared representations by introducing a structure that is valid for all considered tasks. In particular, we look at dense prediction computer vision problems such as monocular depth estimation, semantic segmentation where each input pixel is associated with a target value, and represent their shared intermediate features in a 3D-aware feature space by leveraging recent advances in 3D modeling and differentiable rendering (Niemeyer et al.,

2020; Mildenhall et al., 2020; Chan et al., 2022; 2023). *Our key intuition is that the physical 3D world affords us inherent and implicit consistency between various computer vision tasks.* Hence, by projecting high-dimensional features to a structured 3D-aware space, our method eliminates multiple geometrically-inconsistent cross-task correlations.

To this end, we propose a novel regularization method that can be plugged into diverse prior MTL architectures for dense vision problems including both convolutional (Vandenhende et al., 2020b) and transformer (Ye & Xu, 2022) networks. Prior MTL architectures are typically composed of a shared feature extractor (encoder) and multiple task-specific decoders. Our regularizer, instantiated as a deep network, connects to the output of the shared feature encoder, maps the encodings to three groups of feature maps and further uses these to construct a tri-plane representing planes $x{-}y$, $x{-}z$, $y{-}z$ as in Chan et al. (2022). We are able to query any 3D position by projecting it onto the tri-plane and retrieve a corresponding feature vector through bi-linear interpolation across the planes, passing them through light-weight, task-specific decoders and then rendering the outputs as predictions for each task by raycasting, as in Mildenhall et al. (2020). Once the model has been optimized by minimizing each task loss for both the base model and regularizer, the regularizer is removed. Hence our method does not bring any additional inference cost. Importantly, the regularizer does not require multiple views for each scene and learns 3D-aware representations from a single view. Additionally, our model generalizes to unseen scenes, as the feature encoder is shared across different scenes.

Our method relates to both MTL and 3D modeling work. It is orthogonal to recent MTL contributions that focus rather on designing various cross-task interfaces (Vandenhende et al., 2020a; Liu et al., 2019), or optimization strategies that may obtain more balanced performance across tasks (Kendall et al., 2018; Chen et al., 2018). Alternatively, our main focus is to learn better MTL representations by enforcing 3D structure upon them, through our 3D-aware regularizer. Our method can be incorporated to several recent MTL methods and improve their performance. Most related to ours, Zhi et al. (2021) and Kundu et al. (2022) extend the well-known neural radiance field (NeRF) (Mildenhall et al., 2020) to semantic segmentation and panoptic 3D scene reconstruction, respectively. Unlike them, our main focus is to jointly perform multiple tasks that include depth estimation, boundary detection, surface normal estimation, in addition to semantic segmentation. Uniquely, our method does not require multiple views. Finally, our method is not scene-specific, can learn multiple scenes in a single model and generalizes to unseen scenes.

To summarize, our main contribution is a novel 3D-aware regularization method for the MTL of computer vision problems. Our method is architecture agnostic, does not bring any additional computational cost for inference, and yet can significantly improve the performance of state-of-the-art MTL models as evidenced under two standard benchmarks; NYUv2 and PASCAL-Context.

2 Related Work

Multi-task Learning MTL (Caruana, 1997) commonly aims to learn a single model that can accurately generate predictions for multiple desired tasks, given an input (see Fig. 1 (a)). We refer to Ruder (2017); Zhang & Yang (2017); Vandenhende et al. (2021) for comprehensive literature review. The prior works in computer vision problems can be broadly divided into two groups. The first group (Kokkinos, 2017; Ruder et al., 2019; Vandenhende et al., 2020a; Liang et al., 2018; Bragman et al., 2019; Strezoski et al., 2019; Xu et al., 2018; Zhang et al., 2019; Bruggemann et al., 2021; Bilen & Vedaldi, 2016; Zhang et al., 2018; Xu et al., 2018) focuses on improving network architecture via more effective information sharing across tasks by designing cross-task attention mechanisms Misra et al. (2016), task-specific attention modules (Liu et al., 2019; Bhattacharjee et al., 2023), cross-tasks feature interaction (Ye & Xu, 2022; Vandenhende et al., 2020b), gating strategies or mixture of experts modules (Bruggemann et al., 2020; Guo et al., 2020; Chen et al., 2023; Fan et al., 2022; Ye & Xu, 2023b), visual prompting (Ye & Xu, 2023a; Liu et al., 2023). The second group aims to address the unbalanced optimization for joint minimization of multiple task-specific loss functions, where each may exhibit varying characteristics. This is achieved through either actively changing loss term weights (Kendall et al., 2018; Liu et al., 2019; Guo et al., 2018; Chen et al., 2018; Lin et al., 2019; Sener & Koltun, 2018; Liu et al., 2021b), modifying the gradients of loss functions w.r.t. shared network weights to alleviate task conflicts (Yu et al., 2020; Liu et al., 2021a; Chen et al., 2020; Chennupati et al., 2019; Suteu & Guo, 2019) or knowledge distillation (Li & Bilen, 2020; Li

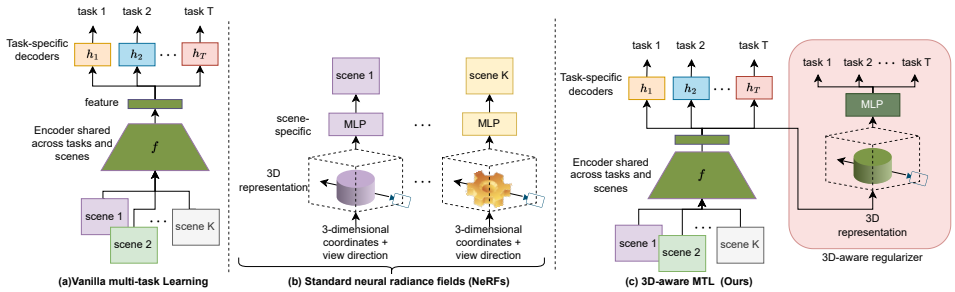

Figure 1: Illustration of (a) vanilla multi-task learning, (b) standard neural radiance fields (NeRFs) and (c) our 3D-aware multi-task learning method.

et al., 2022b). Unlike these methods, our work aims to improve MTL performance by regularizing DNNs through the introduction of 3D-aware representations (see Fig. 1 (c)).

Neural Rendering Our approach also relates to the line of work that learns a 3D scene representations for performing novel view synthesis (Lombardi et al., 2019; Meshry et al., 2019; Sitzmann et al., 2019; Thies et al., 2019; Mildenhall et al., 2020; Chan et al., 2021; Cai et al., 2022; Gu et al., 2021), creating explicit 3D object (Lin et al., 2023), view editing (Zhang et al., 2023). Prior methods with few exceptions can represent only a single scene per model, require many calibrated views, or are not able to perform other tasks than novel view synthesis such as semantic segmentation, depth estimation (see Fig. 1 (b)). PixelNeRF (Yu et al., 2021) conditions a NeRF (Mildenhall et al., 2020) on image inputs through an encoder, allows for the modeling of multiple scenes jointly and generalizes to unseen scenes, however, the work focuses only on synthesizing novel views. Zhi et al. (2021) extend the standard NeRF pipeline through a parallel semantic segmentation branch to jointly encode semantic information of the 3D scene, and obtain 2D segmentations by rendering the scene for a given view using raycasting. However, their model is scene-specific and does not generalize to unseen scenes. Panoptic Neural Fields (Kundu et al., 2022) predict a radiance field that represents the color, density, instance and category label of any 3D point in a scene through the combination of multiple encoders for both background and each object instance. The work was designed for predicting those tasks only on novel views of previously seen scenes, hence it cannot be applied to new scenes without further training on them and is also limited to handle only rigid objects (*c.f.* non-rigid, deformable). In contrast, our method can be used to efficiently predict multiple tasks in novel scenes, without any such restrictions on object type, can be trained from a single view and is further not limited to a fixed architecture or specific set of tasks. Finally, our work harnesses efficient triplane 3D representations from (Chan et al., 2022) that is originally designed to generate high-quality, 3D-aware representations from a collection of single-view images. Our method alternatively focuses on the joint learning of dense vision problems and leverages 3D understanding to bring a beneficial structure to the learned representations.

3 METHOD

We next briefly review the problem settings for MTL and neural rendering to provide required background and then proceed to describe our proposed method.

3.1 MULTI-TASK LEARNING

Our goal is to learn a model $\hat{y}$ that takes in an RGB image $\boldsymbol{I}$ as input and jointly predicts ground-truth labels $Y = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T\}$ for T tasks. In this paper, we focus on dense prediction problems such as semantic segmentation, depth estimation where input image and labels have the same dimensionality. While it is possible to learn an independent model for each task, a more efficient design involves sharing a large portion of the computation across the tasks, via a common feature encoder f. Encoder f takes in an image as input and outputs a high-dimensional feature map which has smaller width and height than the input. In this setting, the encoder is followed by multiple task-specific decoders h_t that each ingests $f(\boldsymbol{I})$ to predict corresponding task labels *i.e.*, $h_t(f(\boldsymbol{I}))$, as depicted in Fig. 1 (a).

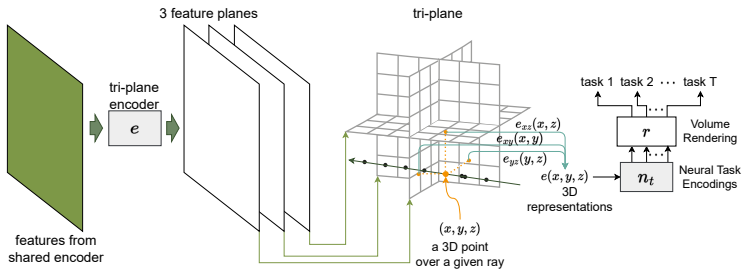

Figure 2: Diagram of 3D-aware regularizer g. The regularizer g takes as input the features from the shared encoder and transforms it to a tri-plane using a tri-plane encoder e. Given a 3D point (x, y, z) on a given ray, we project the coordinates onto three planes and aggregate features from three planes using summation to obtain the 3D representations, which are then fed into a light-weight MLP n_t to estimate predictions of each task or the density of the 3D point. Finally, in volume rendering r, we integrate the predictions over the ray to render the predictions of each task.

Given a labeled training set $\mathcal{D}$ with N image-label pairs, the model weights can be optimized as:

$$\min_{f,\{h_t\}_{t=1}^{T}} \frac{1}{N} \sum_{(\boldsymbol{I},Y)\in\mathcal{D}} \sum_{\boldsymbol{y}_t\in Y} \mathcal{L}_t(h_t \circ f(\boldsymbol{I}), \boldsymbol{y}_t), \tag{1}$$

where $\mathcal{L}_t$ is the loss function for task t, *i.e.*, cross entropy loss for semantic segmentation, L_1 loss for depth estimation. We provide more details in Sec. 4,

3.2 3D-Aware Multi-task Learning

An ideal feature extractor f is expected to extract both task-agnostic and task-specific information, towards enabling the following task-specific decoders to solve their respective target tasks accurately. However, in practice, the combination of high-dimensional feature space and highly non-linear mappings from input to output is prone to overfitting to data and learning of noisy correlations. To mitigate these issues, we propose a new 3D-aware regularization technique that first maps extracted features to 3D neural codes, projects them to task-specific fields and finally renders them to obtain predictions for each target task through differentiable rendering (see Fig. 2). In the regularization, outputs for all tasks are conditioned on observations that lie on a low-D manifold (the density (Mildenhall et al., 2020)), enforcing 3D consistency between tasks.

3D representations. Training the recent MTL models (*e.g.* Vandenhende et al. (2020b); Ye & Xu (2022)) on high resolution input images for multiple dense prediction tasks simultaneously is computation and memory intensive. Hence, naively mapping their multi-scale high-resolution features to 3D is not feasible due to memory limitations in many standard GPUs. Hence, we adopt the hybrid explicit-implicit tri-plane representations of Chan et al. (2022). In particular, we first feed $\boldsymbol{I}$ into a shared encoder and obtain a $W\times H\times C$-dimensional feature map where H and W are the height and width. Then, through a tri-plane encoder e, we project the feature map to three explicit $W\times H\times C'$ dimensional feature maps, e_{xy}, e_{yz}, e_{xz}, that represent axis aligned orthogonal feature planes. We can query any 3D coordinate (x, y, z) by projecting it onto each plane, then retrieve the respective features from three planes via bi-linear interpolation and finally aggregate features using summation to obtain the 3D representation ($e(x, y, z) = e_{xy}(x, y)+e_{yz}(y, z)+e_{xz}(x, z)$) as in Chan et al. (2022).

Neural task fields. For each task, we use an additional light-weight network n_t, implemented as a small MLP, to estimate both a density value and task-specific vector, where this element pair can be denoted as a neural task field for the aggregated 3D representation. We are then able to render these quantities via neural volume rendering Max (1995); Mildenhall et al. (2020) through a differentiable renderer r to obtain predictions for each task. In particular, for the tasks including semantic segmentation, part segmentation, surface normal estimation, boundary detection, saliency prediction, we estimate prediction for each point of a given ray (*e.g.* logits for segmentation) and integrate them over the ray. We normalize the predictions after rendering for surface normal and apply softmax after rendering for segmentation tasks. For depth estimation task, we use the raw prediction as depth maps.

The sequence of mappings can be summarized as: firstly mapping the shared feature encoding $f(\boldsymbol{I})$ to tri-plane features through e, further mapping it to neural task fields through n_t, finally rendering these to obtain predictions for task t, *i.e.* $g_t \circ f(\boldsymbol{I})$ where $g_t = r \circ n_t \circ e$ is the regularizer for task t.

Discussion. While novel view synthesis methods such as NeRF require the presence of multiple views and knowledge of the camera matrices, here we assume a single view to extract the corresponding 3D representations and to render them as task predictions. For rendering, we assume that the camera is orthogonal to image center here, and depict r as a function that takes only the output of n_t but not the viewpoint as input. In the experiments, we show that our model consistently improves the MTL performance, even when learned from a single view per scene, thanks to the 3D structure of representations imposed by our regularizer.

Optimization. We measure the mismatch between ground-truth labels and the predictions obtained from our 3D-aware model branch, and use this signal to jointly optimize the model along with the original common task losses found in Eq. (1):

$$\min_{f,\{h_t,g_t\}_{t=1}^{T}} \frac{1}{N} \sum_{(\boldsymbol{I},Y)\in\mathcal{D}} \sum_{\boldsymbol{y}_t\in Y} \mathcal{L}_t(h_t \circ f(\boldsymbol{I}), \boldsymbol{y}_t) + \alpha_t \underbrace{\mathcal{L}_t(g_t \circ f(\boldsymbol{I}), \boldsymbol{y}_t)}_{\text{3D-aware regularizer}}, \tag{2}$$

where α_t is a hyperparameter balancing loss terms.

Cross-view consistency. Though our 3D-aware regularizer does not require multiple views of the same scene to be presented, it can be easily extended to penalize the cross-view inconsistency on the predictions when multiple views of the same scene are available, *e.g.* video frames. Given two views of a scene $\boldsymbol{I}$ and $\boldsymbol{I}'$ with their camera respective viewpoints V and V', we compute predictions for $\boldsymbol{I}'$ but by using $\boldsymbol{I}$ as the input and render it by using the relative camera transformation ΔV from V to V'. Then we further penalize the inconsistency between this prediction and ground-truth labels of $\boldsymbol{I}'$:

$$\min_{f,\{h_t,g_t\}_{t=1}^{T}} \frac{1}{N} \sum_{\substack{\{(\boldsymbol{I},Y),\\(\boldsymbol{I}',Y')\}\in\mathcal{D}}} \sum_{\substack{\boldsymbol{y}_t\in Y,\\ \boldsymbol{y}'_t\in Y'}} \mathcal{L}_t(h_t \circ f(\boldsymbol{I}), \boldsymbol{y}_t) + \alpha_t \underbrace{\mathcal{L}_t(g_t \circ f(\boldsymbol{I}), \boldsymbol{y}_t)}_{\text{3D-aware regularizer}} + \alpha'_t \underbrace{\mathcal{L}_t(g_t^{\Delta V} \circ f(\boldsymbol{I}), \boldsymbol{y}'_t)}_{\text{cross-view regularizer}}, \tag{3}$$

where α_t and α'_t are hyperparameters balancing loss terms. We set $\alpha_t = \alpha'_t$. Note that in this case $g_t^{\Delta V}$ is a function of ΔV, as the relative viewpoint ΔV is used by the renderer r.

4 EXPERIMENTS

Here we first describe the benchmarks used and our implementation details, then present a quantitative and qualitative analysis of our method.

4.1 DATASET

NYUv2 (Silberman et al., 2012): It contains 1449 RGB-D images, sampled from video sequences from a variety of indoor scenes, which we use to perform four tasks; namely 40-class semantic segmentation, depth estimation, surface normal estimation and boundary detection in common with prior work (Ye & Xu, 2022; Bruggemann et al., 2021). Following the previous studies, we use the true depth data recorded by the Microsoft Kinect and surface normals provided in the prior work (Eigen & Fergus, 2015) for depth estimation and surface normal estimation tasks.

NYUv2 video frames: In addition to the standard data split, NYUv2 (Silberman et al., 2012) also provides additional video frames[1] which are labeled only for depth estimation. Only for the cross-view consistent regularization experiments, we merge the original split with video frames, and train MTL models by minimizing loss on available labeled tasks, *i.e.* all four tasks on the original data and only the depth on video frames. To estimate the relative camera pose ΔV between the frames, we use COLMAP (Schönberger & Frahm, 2016; Schönberger et al., 2016).

[1] https://www.kaggle.com/datasets/soumikrakshit/nyu-depth-v2

PASCAL-Context (Chen et al., 2014): PASCAL (Everingham et al., 2010) is a commonly used image benchmark for dense prediction tasks. We use the data splits from PASCAL-Context (Chen et al., 2014) which has annotations for semantic segmentation, human part segmentation and semantic edge detection. Additionaly, following (Vandenhende et al., 2021; Ye & Xu, 2022), we also consider surface normal prediction and saliency detection using the annotations provided by Vandenhende et al. (2021).

4.2 Implementation Details

Our regularizer is architecture agnostic and can be applied to different architectures. In our experiments, it is incorporated into two state-of-the-art (SotA) MTL methods; MTI-Net (Vandenhende et al., 2020b) and InvPT (Ye & Xu, 2022) which builds on the convolutional neural network (CNN), HRNet-48 (Wang et al., 2020) and transformer based ViT-L (Dosovitskiy et al., 2020) respectively. In all experiments, we follow identical training, evaluation protocols (Ye & Xu, 2022). We append our 3D-aware regularizer to these two models using two convolutional layers, followed by BatchNorm and ReLU, to project feature maps to the tri-plane space resulting in a common size and channel width (64). A 2-layer MLP is used to render each task as in Chan et al. (2022). We use identical hyper-parameters; learning rate, batch size, loss weights, loss functions, pre-trained weights, optimizer, evaluation metrics as MTI-Net and InvPT, respectively. We jointly optimize the task-specific losses and losses arising from our 3D regularization. During inference, the regularizer is discarded. We refer to the supplementary material for further details.

4.3 Results

Comparison with SotA methods. We compare our method with the SotA MTL methods on NYUv2 and PASCAL-Context datasets and report results in Tab. 1 and Tab. 2, respectively. Following Bruggemann et al. (2021), we use HRNet-48 (Wang et al., 2020) as backbone when comparing to CNN based methods; Cross-Stitch (Misra et al., 2016), PAP (Zhang et al., 2019), PSD (Zhou et al., 2020), PAD-Net (Xu et al., 2018), ATRC (Bruggemann et al., 2021), MTI-Net (Vandenhende et al., 2020b). We use ViT-L (Dosovitskiy et al., 2020) as backbone when comparing to InvPT (Ye & Xu, 2022).

In NYUv2 (see Tab. 1), when using HRNet-48 as backbone, we observe that ATRC (Bruggemann et al., 2021) and MTI-Net (Vandenhende et al., 2020b) obtain the best performance. By incorporating our method to MTI-Net (Vandenhende et al., 2020b), we improve its performance on all tasks and outperform all CNN based MTL methods. In comparison, the InvPT approach (Ye & Xu, 2022) achieves superior MTL performance by leveraging both the ViT-L (Dosovitskiy et al., 2020) backbone and multi-scale cross-task interaction modules. Our method is also able to quantitatively improve upon the base InvPT by integrating our proposed 3D-aware regularizer, *e.g.* $+1.31$ mIoU on Seg. The results evidence that the geometric information is beneficial for jointly learning multiple dense prediction tasks.

Method	Backbone	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑
Cross-Stitch (Misra et al., 2016)	HRNet-48	36.34	0.6290	20.88	76.38
PAP (Zhang et al., 2019)	HRNet-48	36.72	0.6178	20.82	76.42
PSD (Zhou et al., 2020)	HRNet-48	36.69	0.6246	20.87	76.42
PAD-Net (Xu et al., 2018)	HRNet-48	36.61	0.6270	20.85	76.38
ATRC (Bruggemann et al., 2021)	HRNet-48	46.33	0.5363	20.18	77.94
MTI-Net (Vandenhende et al., 2020b)	HRNet-48	45.97	0.5365	20.27	77.86
Ours	HRNet-48	**46.67**	**0.5210**	**19.93**	**78.10**
InvPT (Ye & Xu, 2022)	ViT-L	53.56	0.5183	19.04	78.10
Ours	ViT-L	**54.87**	**0.5006**	**18.55**	**78.30**

Table 1: Quantitative comparison of our method to the SotA methods; NYUv2 dataset.

Tab. 2 depicts experimental results on the PASCAL-Context dataset where previous method results are reproduced from Ye & Xu (2022). We also report results from our implementation of the MTI-Net, denoted by 'MTI-Net*', where we found that our implementation obtains better performance. We observe that the performance of existing methods is better than in the previous NYUv2 experiment (Tab. 1), as PASCAL-Context has significantly more images available for training. From Tab. 2 we observe that our method, incorporating our proposed regularizer to MTI-Net (Vandenhende et al., 2020b), can improve the performance on all tasks with respect to our base MTI-Net implementation, *e.g.* +2.29 mIoU on Seg, and obtains the best performance on most tasks compared with MTL

methods that use the HRNet-48 backbone. As in NYUv2, the InvPT model (Ye & Xu, 2022) achieves better performance on a majority of tasks over existing methods. Our method with InvPT again obtains improvements on all tasks over InvPT, *e.g.* $+1.51$ mIoU on PartSeg and $+1.00$ odsF on Boundary. This result further suggests that our method is effective for enabling the MTL network to learn beneficial geometric cues and that the technique can be incorporated with various MTL methods for comprehensive task performance improvements.

Method	Backbone	Seg. (mIoU) ↑	PartSeg (mIoU) ↑	Sal (maxF) ↑	Normal (mErr) ↓	Boundary (odsF) ↑
ASTMT (Maninis et al., 2019)	HRNet-48	68.00	61.10	65.70	14.70	72.40
PAD-Net (Xu et al., 2018)	HRNet-48	53.60	59.60	65.80	15.30	72.50
MTI-Net (Vandenhende et al., 2020b)	HRNet-48	61.70	60.18	84.78	14.23	70.80
ATRC (Bruggemann et al., 2021)	HRNet-48	62.69	59.42	84.70	14.20	70.96
MTI-Net* (Vandenhende et al., 2020b)	HRNet-48	64.42	64.97	84.56	13.82	74.30
Ours	HRNet-48	**66.71**	**65.20**	**84.59**	**13.71**	**74.50**
InvPT (Ye & Xu, 2022)	ViT-L	79.03	67.61	84.81	14.15	73.00
Ours	ViT-L	**79.53**	**69.12**	**84.94**	**13.53**	**74.00**

Table 2: Quantitative comparison of our method to the SotA methods; PASCAL-Context dataset.

3D regularizer with multiple views. Here we investigate whether learning stronger 3D consistency across multiple views with our regularizer further improves the performance in multiple tasks. To this end, we merge the NYUv2 dataset with the additional video frames possessing only depth annotation and train the base InvPT, our method and our method with cross-view consistency on the merged data. For InvPT, we train the model by minimizing losses over the labeled tasks. We train our method by minimizing both the supervised losses and the 3D-aware regularization loss. We further include the cross-view consistency loss as in Eq. (3).

Method	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑
InvPT (Ye & Xu, 2022).	53.44	0.4927	18.78	77.90
Ours	54.93	0.4879	**18.47**	77.90
Ours with cross-view consistency	**54.99**	**0.4850**	18.52	**78.00**

Table 3: Quantitative comparison of our method on NYUv2 dataset + extra video frames with multiple views.

Results of the three approaches are reported in Tab. 3. Compared with the results in Tab. 1, we can see that including video frames for training improves the performance of InvPT on depth and surface normal tasks while yielding comparable performance on remaining tasks. We also see that our method obtains consistent improvement over the InvPT on four tasks with applying 3D-aware regularization using only a single view. Adding the cross-view consistency loss term to our method, we can observe further performance improvement beyond using only single view samples. This suggests that better 3D geometry learning through multi-view consistency is beneficial, however, the improvements are modest. We argue that coarse 3D scene information obtained from single views can be sufficient to learn more structured and regulate inter-task relations.

We also note that this experimental setting is also related to the recent MTL work (Li et al., 2022a) that can learn from partially annotated data by exploiting cross-task relations. However we here focus on an orthogonal direction and believe our complementary works have scope to be integrated together. We leave this as a promising direction for future work.

Comparison with auxiliary network heads. Prior work suggests that the addition of auxiliary heads performing the same task with identical head architectures yet with different weight initializations can be further helpful to performance (Meyerson & Miikkulainen, 2018). To verify whether the improvements obtained by our regularizer is not due to the additional heads solely but introduced 3D structure, we conduct a comparison with our baseline and report results in Tab. 4. The results show that adding auxiliary heads ('InvPT + Aux. Heads') does not necessarily lead to better performance on all tasks; *e.g.* Seg, whereas our method can be seen to outperform this baseline on all tasks suggesting the benefit of introducing 3D-aware structure across tasks.

Method	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑
InvPT (Ye & Xu, 2022)	53.56	0.5183	19.04	78.10
InvPT + Aux. Heads	52.45	0.5131	18.90	77.60
Ours	**54.87**	**0.5006**	**18.55**	**78.30**

Table 4: Quantitative comparison of our method to the baseline of adding auxiliary heads to InvPT; NYUv2 dataset.

3D-aware regularizer predictions. Though we discard the regularizer during inference, the regularizer can also be used to produce predictions for the tasks. To investigate their estimation utility, we report task performance using the default task specific heads h_t, the regularizer output (*regularizer*) and finally using the averaged predictions over two in Tab. 5. We observe that the regularizer alone estimations are worse than the task-specific heads, however, the performance of their averaged output yields marginal improvements to the boundary detection task. The lower performance of using the regularizer alone may be explained by the fact that the rendering image size is typically small (*e.g.* we render 56×72 images for NYUv2). The addition of a super-resolution module, similar to previous work (Chan et al., 2022), can further improve the quality of the related predictions. We leave this to future work.

outputs	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑
task-specific heads	**54.87**	**0.5006**	**18.55**	78.30
regularizer	51.79	0.5282	18.90	74.80
avg	54.68	0.5062	18.70	**78.50**

Table 5: Quantitative results of the predictions from the task-specific heads, regularizer or the average of both the task-specific heads and regularizer in our method; NYUv2 dataset.

Tasks for 3D-aware regularizer. Our regularizer renders predictions for all learning tasks by default. We further study the effect of isolating different tasks for rendering with the regularizer in Tab. 6. Specifically; we jointly optimize the MTL network with a regularizer that renders only one individual task predictions. From Tab. 6 we observe that rendering different individual tasks in the regularizer leads to only marginally differing results and yet using all tasks for rendering can help to better learn the geometric information for MTL, *i.e.* 'All tasks' obtains the best performance on the majority of tasks.

render tasks	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑
Seg.	**55.04**	0.5038	18.78	77.90
Depth	54.62	0.5041	18.93	77.50
Normal	53.43	0.5117	18.59	77.50
Edge	53.97	0.5022	18.97	77.50
All tasks	54.87	**0.5006**	**18.55**	**78.30**

Table 6: Quantitative results of our method isolating different tasks for rendering with the regularizer; NYUv2 dataset.

Using less data. We further investigate the performance gain obtained by our method when trained with fewer training samples. To this end, we train the baseline InvPT (Ye & Xu, 2022) and our method on 25% and 50% of the NYUv2 data after randomly subsampling the original training set. The results are reported in Tab. 7. As expected, more training samples result in better performance in all cases. Our method consistently outperforms the baseline on all tasks in all label regimes with higher margins when more data is available. As the full NYUv2 training set is relatively small, contains only 795 images, our regularizer learns better 3D consistency across tasks from more data too, hence resulting enhanced task performance.

# images	Method	Seg. (mIoU) ↑	Depth (RMSE) ↓	Normal (mErr) ↓	Boundary (odsF) ↑
795 (100%)	InvPT (Ye & Xu, 2022)	53.56	0.5183	19.04	78.10
	Ours	**54.87**	**0.5006**	**18.55**	**78.30**
397 (50%)	InvPT (Ye & Xu, 2022)	49.24	0.5741	20.60	74.90
	Ours	**49.30**	**0.5656**	**20.30**	**76.50**
198 (25%)	InvPT (Ye & Xu, 2022)	43.83	0.6060	21.76	74.80
	Ours	**44.79**	**0.5972**	**21.57**	**74.80**

Table 7: Quantitative comparison of the baseline InvPT and our method in the NYUv2 dataset for varying training set sizes.

4.4 Qualitative Results

We visualize the task predictions for both our method and the base InvPT method on an NYUv2 sample in Fig. 3. Our method can be observed to estimate better predictions consistently for four tasks. For example, our method estimates more accurate predictions around the boundary of the refrigerator, stove and less noisy predictions within objects like curtain and stove. The geometric information learned in our method helps distinguish different adjacent objects, avoids noisy predictions within

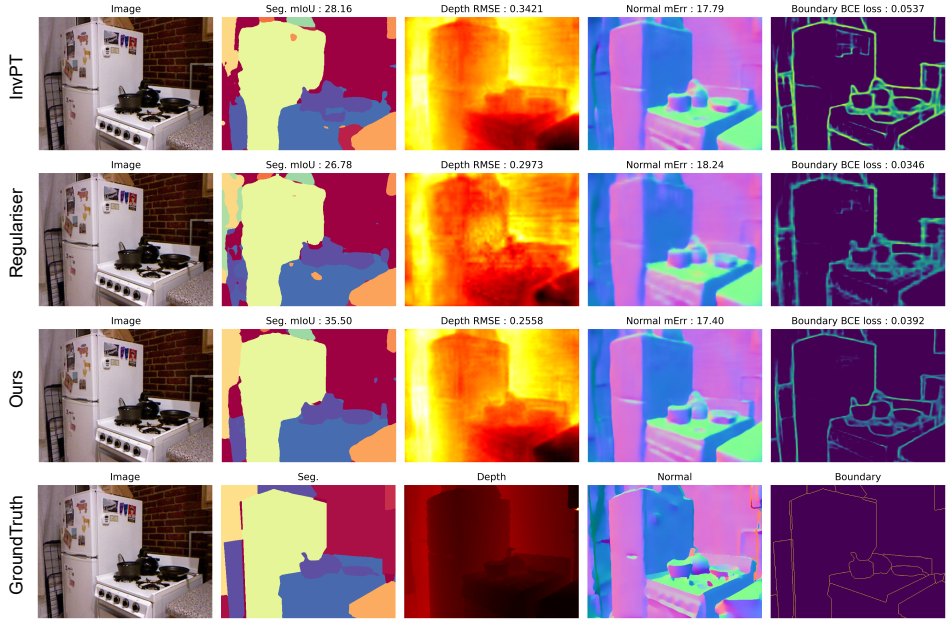

Figure 3: Qualitative results on NYUv2. Each column shows the image or predictions and performance for each task. The last row shows the ground-truth of four tasks. The first to the third row shows the predictions of InvPT, the regularizer in our method and task-specific decoders of our method, respectively.

object boundaries and also improves the consistency across tasks as in the regularizer, all tasks predictions are rendered based on the same density.

We then visualize the predictions of our method's regularizer and the task-specific decoder NYUv2 in Fig. 3. As shown in the figure, our regularizer can also render high quality predictions for different tasks yet it was observed to obtain worse quantitative performance than the task-specific decoders. As discussed, this is due to the rendering image size being usually small (*e.g.* we render 56×72 images for NYUv2).

5 Conclusion and Limitations

We demonstrate that encouraging 3D-aware interfaces between different related tasks including depth estimation, semantic segmentation and surface normal estimation consistently improves the multi-task performance when incorporated to the recent MTL techniques in two standard dense prediction benchmarks. Our model can be successfully used with different backbone architectures and does not bring any additional inference costs. Our method has limitations too. Despite the efficient 3D modeling through the triplane encodings, representing 3D representations for higher resolution 3D volumes is still expensive in terms of memory or computational cost. Rendering specular objects will require different rendering or objects with high frequency 3D details may require more accurate 3D modeling. Furthermore, we balance loss functions with fixed cross-validated hyperparameters, while it would be more beneficial to use adaptive loss balancing strategies (Kendall et al., 2018) or discarding conflicting gradients (Liu et al., 2021a). Finally, in the cross-view consistency experiments where only a portion of images are labeled for all the tasks, our method does not make use of semi-supervised learning or view-consistency for the tasks with missing labels which can further improve the performance of our model.

Acknowledgement. HB is supported by the EPSRC programme grant Visual AI EP/T028572/1. We thank Octave Mariotti, Changjian Li, and Titas Anciukevicius for their valuable feedback.

REFERENCES

Deblina Bhattacharjee, Sabine Süsstrunk, and Mathieu Salzmann. Vision transformer adapters for generalizable multitask learning. *arXiv preprint arXiv:2308.12372*, 2023.

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

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

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

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

Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2nerf: Unsupervised conditional p-gan for single image to neural radiance fields translation. In *CVPR*, pp. 3981–3990, 2022.

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

Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, pp. 5799–5809, 2021.

Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, 2022.

Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. In *CVPR*, 2023.

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

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, pp. 794–803. PMLR, 2018.

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

Zitian Chen, Yikang Shen, Mingyu Ding, Zhenfang Chen, Hengshuang Zhao, Erik G Learned-Miller, and Chuang Gan. Mod-squad: Designing mixtures of experts as modular multi-task learners. In *CVPR*, pp. 11828–11837, 2023.

Sumanth Chennupati, Ganesh Sistu, Senthil Yogamani, and Samir A Rawashdeh. Multinet++: Multi-stream feature aggregation and geometric loss strategy for multi-task learning. In *CVPR Workshop*, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

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

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

Zhiwen Fan, Rishov Sarkar, Ziyu Jiang, Tianlong Chen, Kai Zou, Yu Cheng, Cong Hao, Zhangyang Wang, et al. M^3vit: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design. *Neurips*, 35:28441–28457, 2022.

Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. *arXiv preprint arXiv:2110.08985*, 2021.

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

Pengsheng Guo, Chen-Yu Lee, and Daniel Ulbricht. Learning to branch for multi-task learning. In *ICML*, pp. 3854–3863. PMLR, 2020.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pp. 7482–7491, 2018.

Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, pp. 6129–6138, 2017.

Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas J Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic neural fields: A semantic object-aware neural scene representation. In *CVPR*, pp. 12871–12881, 2022.

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

Wei-Hong Li, Xialei Liu, and Hakan Bilen. Learning multiple dense prediction tasks from partially annotated data. In *CVPR*, 2022a.

Wei-Hong Li, Xialei Liu, and Hakan Bilen. Universal representations: A unified look at multiple task and domain learning. *arXiv preprint arXiv:2204.02744*, 2022b.

Jason Liang, Elliot Meyerson, and Risto Miikkulainen. Evolutionary architecture search for deep multitask networks. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 466–473, 2018.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *CVPR*, pp. 300–309, 2023.

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

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *NeurIPS*, 2021a.

Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *ICLR*, 2021b.

Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *CVPR*, pp. 1871–1880, 2019.

Yajing Liu, Yuning Lu, Hao Liu, Yaozu An, Zhuoran Xu, Zhuokun Yao, Baofeng Zhang, Zhiwei Xiong, and Chenguang Gui. Hierarchical prompt learning for multi-task learning. In *CVPR*, pp. 10888–10898, 2023.

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 2019.

Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *CVPR*, pp. 1851–1860, 2019.

Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995.

Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *CVPR*, pp. 6878–6887, 2019.

Elliot Meyerson and Risto Miikkulainen. Pseudo-task augmentation: From deep multitask learning to intratask sharing—and back. In *ICML*, pp. 3511–3520. PMLR, 2018.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, pp. 3994–4003, 2016.

Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, pp. 3504–3515, 2020.

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

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *AAAI*, volume 33, pp. 4822–4829, 2019.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *NeurIPS*, 2018.

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

Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *CVPR*, pp. 2437–2446, 2019.

Gjorgji Strezoski, Nanne van Noord, and Marcel Worring. Many task learning with task routing. In *ICCV*, pp. 1375–1384, 2019.

Mihai Suteu and Yike Guo. Regularizing deep multi-task networks using orthogonal gradients. *arXiv preprint arXiv:1912.06844*, 2019.

Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *TOG*, 38(4):1–12, 2019.

Simon Vandenhende, Stamatios Georgoulis, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: deciding what layers to share. In *bmvc*, 2020a.

Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *ECCV*, pp. 527–543. Springer, 2020b.

Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *PAMI*, 2021.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *TPAMI*, 43(10):3349–3364, 2020.

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*, pp. 675–684, 2018.

Hanrong Ye and Dan Xu. Inverted pyramid multi-task transformer for dense scene understanding. In *ECCV*, pp. 514–530. Springer, 2022.

Hanrong Ye and Dan Xu. Taskprompter: Spatial-channel multi-task prompting for dense scene understanding. In *ICLR*, 2023a.

Hanrong Ye and Dan Xu. Taskexpert: Dynamically assembling multi-task representations with memorial mixture-of-experts. In *ICCV*, pp. 21828–21837, 2023b.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, pp. 4578–4587, 2021.

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

Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao. Text2nerf: Text-driven 3d scene generation with neural radiance fields. *arXiv preprint arXiv:2305.11588*, 2023.

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

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

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*, pp. 4106–4115, 2019.

Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, pp. 15838–15847, 2021.

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