Vision Transformer Adapters for Generalizable Multitask Learning

Deblina Bhattacharjee, Sabine Süsstrunk and Mathieu Salzmann
School of Computer and Communication Sciences, EPFL, Switzerland
{deblina.bhattacharjee, sabine.susstrunk, mathieu.salzmann}@epfl.ch

Abstract

We introduce the first multitasking vision transformer adapters that learn generalizable task affinities which can be applied to novel tasks and domains. Integrated into an off-the-shelf vision transformer backbone, our adapters can simultaneously solve multiple dense vision tasks in a parameter-efficient manner, unlike existing multitasking transformers that are parametrically expensive. In contrast to concurrent methods, we do not require retraining or fine-tuning whenever a new task or domain is added. We introduce a task-adapted attention mechanism within our adapter framework that combines gradient-based task similarities with attention-based ones. The learned task affinities generalize to the following settings: zero-shot task transfer, unsupervised domain adaptation, and generalization without fine-tuning to novel domains. We demonstrate that our approach outperforms not only the existing convolutional neural network-based multitasking methods but also the vision transformer-based ones. Our project page is at https://ivrl.github.io/VTAGML.

1. Introduction

In the past few years, vision transformers [5, 15, 16, 26, 30, 57] have grown in popularity at an incredible pace. They have now achieved state-of-the-art results, outperforming Convolutional Neural Network (CNN) based methods not only in image classification [22, 23, 26] but also in many dense prediction tasks such as semantic segmentation [11, 47, 59, 68], monocular depth estimation [40, 61], and surface normal prediction [24, 62]. Therefore, utilizing the power of vision transformers in a unified framework to simultaneously solve multiple tasks seems a natural way forward. Nevertheless, only a few works [3, 7, 21, 35, 44] have attempted this so far, and all of them rely on handcrafted transformer architecture designs. Specifically, IPT and ST-MTL [7, 35] exploit a multi-head multi-tail architecture tailored to solve specific tasks; MulT [3] leverages a pairwise task attention strategy handcrafted to utilize surface normal prediction as reference task for dense predictions; and UniT [21] as well as Vid-MTL [44] use a multimodal transformer architecture to achieve multiple pairwise task predictions across different modalities. While these multitasking vision transformer-based methods [7, 3, 21, 35, 44] outperform their multitasking CNN-based counterparts [29, 31, 34, 45, 48, 52, 53, 58, 65, 66], none of the existing vision transformer-based or CNN-based MTL methods can adapt to new tasks as well as to novel domains. In fact, it was observed in the seminal work of [66] and confirmed in subsequent MTL studies [3, 45, 65] that the multitask affinities learned by existing MTL frameworks are *not* transferable or generalizable.

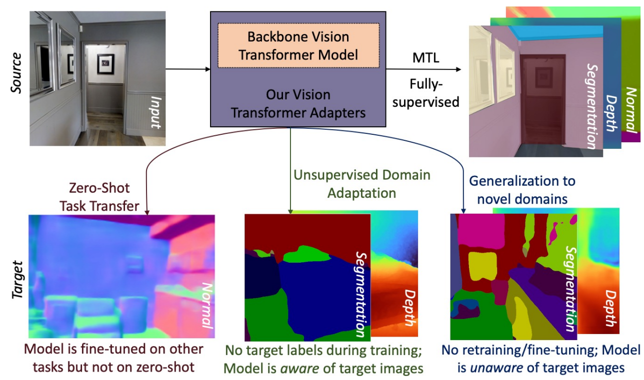

Figure 1: **Motivation of our work.** Unlike existing MTL methods, our vision transformer adapters generalize to novel tasks and domains.

This raises the following question: Is there a way we can learn transferable and generalizable task affinities such that multitask affinities transfer to novel tasks and generalize to novel domains, thereby allowing us to reuse an existing network? To answer this, we introduce vision transformer adapters for generalizable multitask learning and propose an *automated* framework that can learn *transferable and generalizable* task affinities which can adapt to new tasks or domain representations in a *parameter-efficient* manner. Additionally, unlike existing transformer-based handcrafted MTL methods [7, 3, 21] that learn task affinities in a pairwise manner, our vision transformer adapters learn task affinities in an automated way and across *all* the tasks.

To achieve this, we equip our vision adapters with three mechanisms: (1) An improved gradient-based task similarity approach (TROA) first introduced in [8]; (2) a novel task-adapted attention mechanism (TAA) that combines the gradient-based task similarities with attention-based ones, thereby learning transferable and generalizable task affinities; and (3) a task-scaled normalization to account for the different task scales. The resulting module can then be seamlessly integrated with a pre-trained, frozen encoder backbone architecture such as ViT [16], Swin [30], Pyramid Transformer [56], or Focal Transformer [63]. Our approach is independent of the choice of the vision transformer backbone, unlike existing transformer-based MTL methods. Our contributions are summarized as follows:

- We introduce vision adapters for generalizable multitask learning that leverages a pre-trained vision transformer backbone to learn transferable and generalizable features at a low computational cost.
- At the heart of our vision adapter, we introduce a novel task-adapted attention mechanism (TAA) that automatically learns task dependencies from the shared representation, by combining gradient-based task similarities (TROA) with attention-based ones.
- Our task affinities transfer to different settings including multitask learning, zero-shot task transfer learning, and unsupervised domain adaptation. Moreover, our task affinities generalize to novel domains *without* requiring any fine-tuning.
- Our multitasking vision transformer adapters can be integrated with different transformer backbones such as ViT [16], Swin [30], Pyramid Transformer [56], and Focal Transformer [63], achieving a significant increase in performance in a parameter-efficient way.

Our experiments evidence that our method outperforms both state-of-the-art CNN-based multitasking methods [8, 29, 31, 34, 37, 48, 52, 65, 66] as well as transformer-based ones [3, 21].

2. Related Work

Multitask Learning. Multitask learning has been a fundamental problem for years; see Vandenhende et. al. [52] for a great survey. As noted by multiple works [18, 45, 52], MTL networks are unstable and require a strong *balance* between tasks to perform well. Prior works [29, 31, 34, 45, 48, 52, 53, 58, 65, 66] aim to strike this balance either using a gradient-based learning of task affinities in the encoded representations [8, 29, 34, 52, 64], or applying task conditioned gates to the decoder [48], attention-based task similarities [31, 53, 58] or weighted task losses [9, 13, 37]. While these works, all based on the convolutional neural network (CNN) backbone, show promising results, they remain challenged by negative task transfer, i.e., the degraded performance of certain tasks when learned jointly. To overcome this, Standley et. al [45] developed subsets of complementary tasks where each of these subsets, when trained, can overcome negative task transfer. Being a handcrafted approach, [45] resulted in a large number of subsets comprising different task combinations.

Following this, IPT [7] was the first transformer-based multitask network aiming to solve low-level vision tasks after fine-tuning a large pre-trained network. Subsequently, [35], jointly addressed the tasks of object detection and semantic segmentation, and [44] used a similar architecture for scene and action understanding in videos. Recently, Hu et.al. [21] proposed a framework that tackles several language tasks but a single vision one. MulT [3] showed the superiority of vision transformers over CNN-based networks in modeling the multitask affinities via its shared attention mechanism, thereby solving all the tasks in a single model. While these transformer-based frameworks [3, 7, 21, 35, 44] clearly outperform the existing CNN-based multitasking methods, they are handcrafted and cannot be integrated into a different transformer backbone. By contrast, our vision transformer adapters can be integrated into an off-the-shelf vision transformer backbone, while learning task affinities based on *all* the tasks in an automated manner.

Learning generalizable task affinities. Taskonomy [67] studied the relationships between multiple visual tasks for transfer learning. Following this, a number of recent works have studied tasks relationships for transfer learning [1, 17, 18, 38, 45, 55]. These works analyze a network that is trained on a source task and is applied to a different target task. None of these mentioned works demonstrate a correlation between the transfer task affinities and the multitask affinities. To address this, we introduce our multitask vision transformer adapters that can successfully transfer the multitask affinities to novel tasks *and* novel domains.

Vision Transformer Adapters. First introduced for language tasks to leverage knowledge embedded in large pretrained transformers, adapters [20] are trainable modules that are attached to specific locations of a pre-trained transformer network, providing a way to limit the number of parameters needed when confronted with a large number of tasks. This approach is also effective with pretrained vision transformers that have rich semantic information [10, 25, 27]. Specifically, Li et al. [25, 27] proposed ViT-based adapters for object detection, whereas Chen et al. [10] added feed-forward bottlenecks in every transformer block for the separate downstream tasks of object detection and semantic segmentation. Such methods, however, adapt to a *single* downstream task. By contrast, we propose vision transformer adapters that can infer on *multiple* dense-vision tasks in a single run in a parameter-efficient manner. To the best of our knowledge, only the

		Architecture				Task-affinity generalization			
	Methods	Encoder-focused	Decoder-focused	Attention	Task-loss	MTL	Task-transfer	UDA	Novel domain
CNN-based	MTL-baseline [52]	✓	✗	✗	✗	✓	✗	✗	✗
	Consistency [65]	✗	✓	✗	✓	✓	✓	✗	✗
	XTAM [31]	✗	✗	✓	✗	✓	✗	✓	✗
	TAWT [8]	✓	✗	✗	✗	✓	✓	✗	✗
	Cross-stitch [34]	✓	✗	✗	✗	✓	✗	✗	✗
	MTAN [29]	✓	✗	✗	✗	✓	✗	✗	✗
	TSwitch [48]	✗	✓	✗	✓	✓	✗	✗	✗
	TTNet [37]	✗	✗	✗	✓	✓	✓	✗	✗
	Taskonomy [67]	✗	✓	✗	✓	✓	✓	✗	✗
Vision Transformer-based	ST-MTL [35]	✗	✓	✓	✓	✓	✗	✗	✗
	MulT [3]	✗	✓	✓	✓	✓	✗	✓	✗
	Our	✓	✗	✓	✓	✓	✓	✓	✓

Table 1: **Taxonomy of MTL approaches.** Our vision transformer adapter method is an encoder-focused, task-balanced approach that uses task-adapted attention (TAA) to learn generalizable task affinities, unlike existing CNN-based and vision transformer-based MTL methods. Here, we list the methods that we evaluate in this work. A detailed taxonomy of other MTL methods is provided in supplementary.

prior works of [46, 49]—both in the field of NLP—mix multitask learning and adapters within large pre-trained *language* transformers by creating local task modules that are controlled by a global task-agnostic module. This approach, however, has the drawback of adding new non-shared parameters whenever a new task is added, thereby failing to generalize on novel tasks. By contrast, our vision transformer adapters share all parameters across the tasks and can re-modulate the existing weights when a new task is introduced. Moreover, the task affinities learned by our vision transformer adapters generalize to novel domains, unlike any existing work.

Transformer Attention Mechanisms. While many works exploit the long-range dependencies of transformers by computing a task-specific attention [6, 12, 56, 57, 60, 63] and pairwise task attention for MTL [3], none of these attention mechanisms learn task-affinities based on *all* the tasks in an automated manner. We, therefore, introduce a task-adapted attention (TAA) mechanism that learns the task affinities by combining gradient-based task similarities with the attention ones. In essence, our TAA conditions the self-attention of the transformer backbone on the gradient-based task similarities.

3. Method

Our novel vision transformer adapter method achieves predictions for a novel task or domain by learning transferable and generalizable task affinities. Our adapters leverage pre-trained vision transformer models that are readily and ubiquitously available. While these easily available vision transformer models are pre-trained for classification on ImageNet, we aim to integrate them with multitasking. This calls for learning multitask affinities. To achieve this, within our vision adapters, we compute the gradient-based task similarity approach (TROA—Section 3.2.1), that is, in turn, used by a novel task-adapted attention mechanism (TAA—Section 3.2.2). This yields representations that are then normalized according to the task scales (Section 3.2.3), and finally decoded by the task-specific decoders and their respective task heads. Our overall framework is shown in Figure 2. Below, we discuss its different modules in detail. Note that, although we present it using the Swin-B architecture, which is the most widely used backbone for dense prediction, our method can be integrated with any existing vision transformer backbone, such as ViT [16], Pyramid Transformer [56] or Focal Transformer [63], as will be shown by our experiments in the supplementary.

3.1. Encoder Module

For the encoder, we adopt a pre-trained Swin-B V2 [30] model initialized with ImageNet-22K-trained weights. The encoder comprises four successive transformer stages employing a patch embedding that gradually decreases the resolution of the input image in a pyramidal manner while increasing the channel dimension. As shown in Figure 2, the first, second, and fourth stages have 2 transformer blocks while the third stage has 18 blocks. That is, following [19], most of the computation is concentrated in the third stage. Therefore, we propose to add trainable vision adapters on top of this stage — specifically for transformer blocks 15 to 18 — to leverage the rich embeddings it extracts. Nonetheless, to further reason about the high-level semantic information encoded in the final representation, we add two vision adapters for both transformer blocks in the fourth Swin stage. For any other vision transformer backbone [16, 56, 63], our vision transformer adapters work best when integrated with layers comprising mid-level to high-level information.

3.2. Vision Transformer Adapter Module

Our vision transformer adapters, depicted in Figure 3, build on a sequence of transformer layers of length consistent with the Swin's inter-window connectivity configurations. We connect the consequent adapter layers by using skip connections where the output of the previous layer is an input to the next layer. This connectivity allows information to flow from preceding layers to later ones. Within each vision adapter, different mechanisms are at play. In

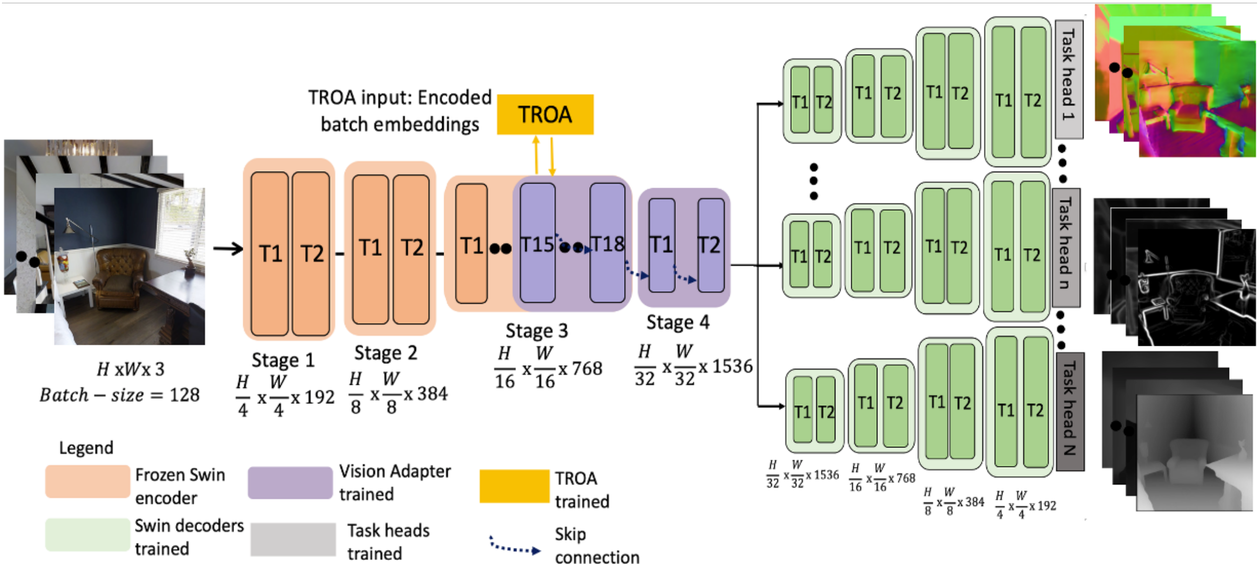

Figure 2: **Detailed overview of our method.** The frozen transformer encoder module (in orange) extracts a shared representation of the input image, which is then utilized to learn the task affinities in our novel vision transformer adapters (in purple). Each adapter layer uses gradient task similarity (TROA) (in yellow) and Task-Adapted Attention (TAA) to learn the task affinities, which are communicated with skip connections (in blue) between consecutive adapter layers. The task embeddings are then decoded by the fully-supervised transformer decoders (in green) for the respective tasks. Note that the transformer decoders are shared but have different task heads (in grey). For clarity, only three tasks are depicted here and TAA is explained in a separate figure.

particular, these mechanisms are (i) TROA, which builds on [8], and optimizes the task representations by computing their gradient similarity; (ii) a novel task-adapted attention (TAA) module to combine gradient-based task affinities from TROA with attention ones; and (iii) a novel task-scaled normalization (TSN) approach to balance the task scales. The adapter framework also relies on a bottleneck network consisting of a linear down-projection (FF down), a non-linearity, and a linear up-projection (FF up), used to decrease the number of parameters. In detail, for a batch of input images, where each image can be denoted as $X \subset \mathbb{R}^{H\times W\times 3}$, the vision adapter encodes the representation of the batch of images as $\hat{\phi}$. These batch embeddings are normalized using a layer norm operation [2]. Once normalized, the embeddings are passed onto our novel TAA module which triggers the TROA mechanism within it to find the task similarities. We now explain the gradient-based task similarity computed by TROA.

3.2.1 Task Representation Optimization Algorithm (TROA)

TROA computes a task representation $\hat{\theta}$ and a task affinity matrix $\hat{\omega}$ that depends on how correlated the tasks are. It is, therefore, named the Task Representation Optimization Algorithm, since it optimizes the task representations based on the task gradients. Specifically, this is computed by a gradient-based task affinity which gives an interpretive

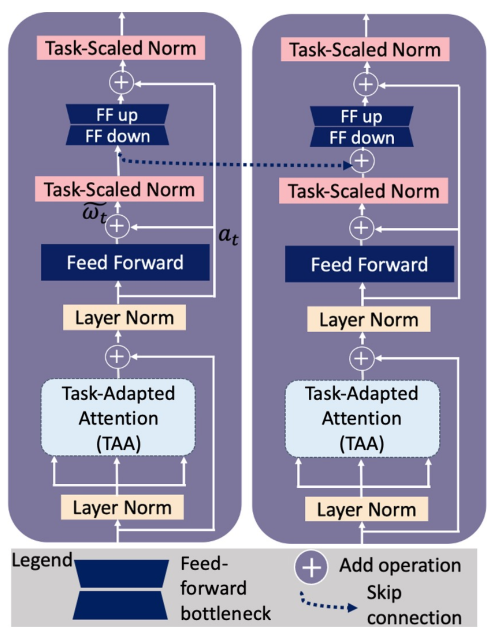

Figure 3: **Overview of our vision transformer adapter module.** Our vision adapters learn transferable and generalizable task affinities in a parameter-efficient way. We show two blocks to depict the skip connectivity between them.

measure of the influence of an inductive task $n \in [1, N]$ on a target task $t \in [1, N]$ based on the similarity between their learned representations. In TROA, we estimate these task affinities as the cosine similarity, dubbed $sim_{t,n}$ in Algorithm 1, between the gradient of the inductive task and the gradient of the target task. This cosine similarity is computed for *all* task combinations. Specifically, at iteration i, TROA starts with $\left(\theta^i, \{m_n^i\}_{n=1}^N\right)$, i.e., the feature representation and the corresponding N task-specific decoder functions. Upon making a forward pass, it learns the task weights by minimizing the overall multitask learning objective described by $\sum_{n=1}^{N} \omega_{t_n}^i \widehat{\mathcal{L}}_n(\theta, m_n)$ via Adam [32]. We then calculate the cosine similarity between the task gradients to, ultimately, compute the task affinities. We employ a closed-form solution for the analytical weight update, w_t^{i+1}, given by the approximate mirror descent formula [51] with a step-size $\kappa = 1$. Note that the task weight vector $\hat{\omega}_t$ is updated via a combination of alternating minimization and mirror descent, where the minimization step prevents mode collapse if the task weights become equal. At the end of the i^{th} iteration, we obtain a new task representation and a new weight vector $\hat{\omega}_t$ for the t^{th} task, identifying its affinity with all the tasks.

Algorithm 1 TROA

Input: Batch embedding from vision adapter $\hat{\phi}$
Output: Task representation $\widehat{\theta}$, task-specific decoder function $\widehat{m_t}$ and weight vector $\widehat{\omega_t}$ for the 't'th task .
Initialize: $\boldsymbol{\omega}_t^1 \in \mathbb{R}^N$ uniformly,
$\theta^1 \leftarrow \hat{\phi}, \{m_n^1\}_{n=1}^N \subset \mathcal{M}$;
for $i = 1, \ldots, I-1$ **do**
  Starting With $\left(\theta^i, \{m_n^i\}_{n=1}^N\right)$; % i^{th} iteration.
  Run a few steps of Adam to minimize $\sum_{n=1}^{N} \omega_{t_n}^i \widehat{\mathcal{L}}_n(\theta, m_n)$ and get $\left(\theta^{i+1}, \{m_n^{i+1}\}_{n=1}^N\right)$;
  Run $sim_{t,n}^i := cossim(\nabla_\theta \widehat{\mathcal{L}}_t(\theta^{i+1}, m_t^{i+1}), \nabla_\theta \widehat{\mathcal{L}}_n(\theta^{i+1}, m_n^{i+1}))$; % gradient similarity.
  Update $\omega_t^{i+1} := \frac{\omega_t^i \exp\{-\kappa sim_{t,n}^i\}}{\sum_{n'=1}^{N} \omega_{n'}^i \exp\{-\kappa sim_{t,n'}^i\}}$;
end for
return $\widehat{\theta} = \theta^I, \widehat{m_t} = m_t^I, \widehat{\omega_t} = \omega_t^I$

In Figure 4 we show the task affinities from TROA when four tasks comprising semantic segmentation (SemSeg), depth, surface normal, and edges are jointly learned. We show that TROA learns a strong task affinity between the same task gradients, for example, segmentation with segmentation. This is a self-explanatory observation. Consequently, TROA also learns task affinities between proximate tasks such as segmentation and depth, and task affinities between non-proximate tasks such as segmentation and normal. Note that task dependence is assymetric, i.e. segmentation does not affect normal as normal effects segmentation. This is evidenced in Figure 4 and also by prior works [3, 18, 45]. These task affinities are used by our novel task-adapted attention module as described in the following section.

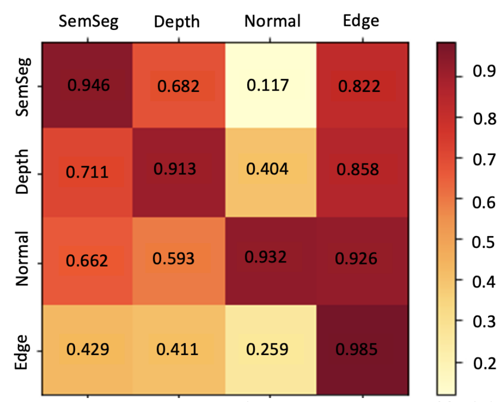

Figure 4: We show the **gradient-based task affinities**, $\hat{\omega} \in \mathbb{R}^{N \times N}$ returned by TROA for N tasks.

3.2.2 Task-Adapted Attention (TAA)

Our task-adapted attention module, as shown in Figure 5, combines gradient-based task affinities, represented by $\hat{\omega}_t$, with attention-based ones, represented by $q \cdot k^T$. The

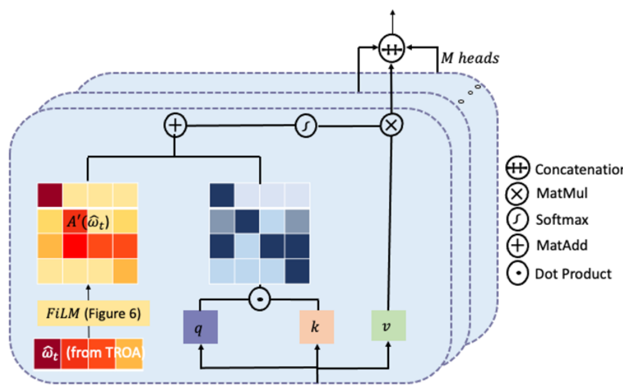

Figure 5: Overview of our **Task-Adapted Attention (TAA)** mechanism that combines task affinities with image attention. Note that the process, in the foreground, is for a single attention head which is repeated for M heads to give us the task-adapted multi-head attention.

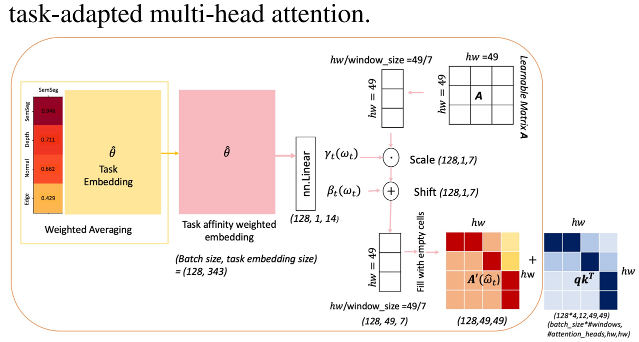

Figure 6: **Detailed overview of Feature Wise Linear Modulation (FiLM)** which linearly shifts and scales tasks representations to match dimensions of the feature maps. The orange rectangular area is FiLM.

gradient-based task affinities, $\hat{\omega_t}$, are obtained from TROA as discussed in Section 3.2.1. In a parallel branch, we extract a query q, key k, and value v matrix from $\hat{\phi}$, following the standard approach in attention-based methods [54]. The widely-known self-attention (SA) [54] is computed as,

$$SA(q, k, v) = softmax[q \cdot k^T / \sqrt{c_{qkv}}]v \ , \tag{1}$$

where c_{qkv} is the channel dimension of the query, key, and value. In contrast to this standard formulation, we condition the self-attention on the gradient-based task affinities, $\hat{\omega_t}$ from TROA (Section 3.2.1). Formally, for a given task t, our task-adapted attention is

$$TAA(q, k, v, \hat{\omega_t}) = softmax[A'(\hat{\omega_t}) + q \cdot k^T / \sqrt{c_{qkv}}]v \ , \tag{2}$$

$$\text{where } A'(\hat{\omega_t}) = A\gamma_t(\hat{\omega_t}) + \beta_t(\hat{\omega_t}) \ . \tag{3}$$

Here, $\hat{\omega_t}$ is the N-dimensional vector of affinities for task t, i.e., the t^{th} column of $\hat{\omega}$. As $\hat{\omega_t} \in \mathbb{R}^N$, we apply the widely-used Feature Wise Linear Modulation [39] to match its dimension to the spatial dimensions of the feature maps and thus get $A'(\hat{\omega_t})$. Specifically, the Feature Wise Linear Modulation (FiLM) [39] performs weighted averaging of the task representations w.r.t the task affinity weights, and then linearly shifts and scales the task representations as seen in Figure 6. It is more stable, unlike other dimension-matching techniques and we use this technique in our TAA module to match the dimensions of the affinity matrix $\hat{\omega_t}$ to the spatial dimensions of the feature maps and thus get $A'(\hat{\omega_t})$.

Formally, as indicated in Eq. 3, $A'(\hat{\omega_t})$ is computed by first mapping $\hat{\omega_t} \in \mathbb{R}^N$ to matrices of size $hw \times hw$ via the Feature Wise Linear Modulation [39] functions γ_t, β_t. The matrix output by $\gamma_t(\hat{\omega}^t)$ is then linearly transformed by a randomly-initialized learnable matrix A. Subsequently, we combine $A'(\cdot)$ with the $q \cdot k^T / \sqrt{c_{qkv}}$ matrix to obtain the TAA as in Eq. 2. In essence, for the t^{th} task, the TAA module aids the query and key matrix to compute attention from the most similar tasks. Note that we generically use h and w to denote the spatial dimensions of the feature maps at different stages (i.e., $H/16, W/16$ for stage 3, and $H/32, W/32$ for stage 4).

The process described above corresponds to a single attention head. In practice, as shown in Figure 5, we perform this for M heads, where $M = 24$ and 48 for the third and fourth stage, respectively, resulting in M task-specific feature vectors. We then concatenate these vectors into a representation. Note that we apply the same procedure for the task-adapted attention in all the vision adapters. We defer qualitative comparisons of our TAA module w.r.t the typical self-attention (SA) to the supplementary.

Referring to Figure 3, the output of the task-adapted multi-head attention is employed in a residual connection followed by a layer norm operation, a feed-forward network, and another residual addition resulting in an $\tilde{\omega_t}$ matrix. This matrix is scaled w.r.t. the task t. Our vision adapters achieve task-scaling by employing the Task-Scaled Norm, which is described in the following section.

3.2.3 Task-Scaled Normalization (TSN)

TSN balances the different scales of the tasks. Balancing the task scales is necessary to avoid learning interference in a multitasking framework [52]. To this end, inspired by the Conditional Batch Normalization [33] strategy, we formulate TSN as follows. For task $t; t \in 1, \ldots, N$,

$$\begin{aligned} TSN_t = \frac{1}{\sigma} * (a_t - \mu) * \hat{\gamma}_t\left(\tilde{\omega}_t\right) + \beta_t\left(\tilde{\omega}_t\right) \ , \\ \text{where } \hat{\gamma}_t(\tilde{\omega}_t) = \gamma' \gamma_t\left(\tilde{\omega}_t\right) + \beta' \ , \end{aligned} \tag{4}$$

a_t, as shown in Figure 3, is the task-specific activation obtained from the residual connection, and $\tilde{\omega_t}$ is the summed output of the feed-forward network with the residual connection. Furthermore, μ and σ are the mean and the variance of all the inputs within each layer, as defined in [2], and γ' and β' are the Swin's Layer Normalization weight and bias functions. Our TSN mechanism contrasts with Layer Norm in the following two ways: 1) While the Layer Norm weight and bias functions are kept fixed, the TSN ones are trained; 2) while Layer Norm normalizes the input across features, TSN modulates the normalization output based on the task weights.

3.3. Decoder Module

Leveraging the idea in [30, 3], our decoder architecture comprises four stages, each containing 2 sequential transformer blocks for a total of 8. In each stage, the two sequential transformer blocks alternate regular and shifted window attention mechanisms, as in [30]. Between each stage, we employ an upsampling layer to double the spatial resolution and halve the channel dimension; we therefore adjust the number of attention heads accordingly to 48, 24, 12, and 6, in the first, second, third, and fourth stage, respectively. Unlike in [3], where the lower-resolution stages of the decoder are guided by the higher-level deeper encoded features and vice versa, our model employs trainable vision adapters to guide the stages of the decoder in a sequential manner. To perform predictions on multiple tasks, we share the vision adapters across all tasks and use task-specific decoders with the same architecture but different parameter values. We then simply append task-specific heads to the decoder.

Task Heads and Training. The decoded feature representations are passed into the linear task-specific heads, such that the task head outputs an $H \times W \times K$ map, where H, W, and K are the input image dimensions and the task-specific channels, respectively. We jointly train the adapters and the decoders by employing a linear combination of the task losses, where the losses are calculated between the ground truth and predictions for each task. To maintain consistency with the baselines [3, 31, 65], we use the cross-entropy for segmentation, the rotated loss for depth, and the l_1 loss for surface normal and 2D edges, respectively.

4. Experiments and Results

Considering the number of experiments and results we report, we highlight in the main paper one consistent set of results and defer additional qualitative and quantitative results to the supplementary material. For easier comparison, we only report here the results of our vision adapters with the SWIN-B transformer backbone. Results with other transformer backbones like ViT [16], Pyramid Transformer [56], and Focal Transformer [63] are also in the supplementary along with descriptions of the datasets, baselines, and evaluation metrics that we use.

Experimental Setup. The experiments were performed using the following 4 dense prediction tasks: semantic segmentation (*S*), depth (zbuffer) (*D*), surface normal (*N*), and 2D (Sobel) texture edges (*E*). We report results in the following settings: 1) The *MultiTask Learning (MTL) setting*; 2) the *Zero-shot Task Transfer setting*; 3) the *Unsupervised Domain Adaptation (UDA) setting*; and 4) *Generalization to Novel Domains*.

For the MTL setting, the methods are jointly trained in a fully-supervised manner on task combinations such as *'S-D'* (segmentation + depth), *'S-D-N'* (segmentation + depth + normal) and *'S-D-N-E'* (segmentation + depth + normal + edges) on the Taskonomy benchmark [67] and the NYUDv2 benchmark [36]. We also evaluate all models on Synthia [42], Cityscapes [14], and Vkitti2 [4] for the task combinations *'S-D'* and *'S-D-N'*.

For the Zero-shot Task Transfer setting, all the methods are first trained on Vkitti2 [4] and then fine-tuned on Cityscapes or Synthia using the ground-truth segmentation labels in the *'S-D'* case, and the ground-truth segmentation and depth labels in the *'S-D-N'* one.

For the UDA setting, we deal with distribution shifts between a source domain, with labeled data, and a target domain, in which only unlabeled data is available for training. All models are trained with the source domain labels of Vkitti2 [4] with the models *aware* of the images in both the source [4] and target [14] domain.

To evaluate the generalizability of our learned task affinities to novel domains, wherein the model is *unaware* of the images in the target domain, we train the models on Taskonomy [67] and apply them to NYUDv2 [36] without any fine-tuning. Furthermore, we train our model on MS-Coco [28] and apply them to a highly disparate comics domain that differs in styles and contents from real-world imagery.

4.1. Qualitative Results

We qualitatively compare the results of our model with different baselines [3, 8, 31, 35] in Figure 7 for the tasks of segmentation, depth, normal and edge prediction on the Taskonomy [67] benchmark for the MTL setting. Our method yields higher-quality predictions than all the baselines. This is noticeable when looking at thin elements (e.g., flower vases, and table lamps) and object contours. The visuals correspond to the quantitative analysis. More qualitative results are provided in the supplementary material.

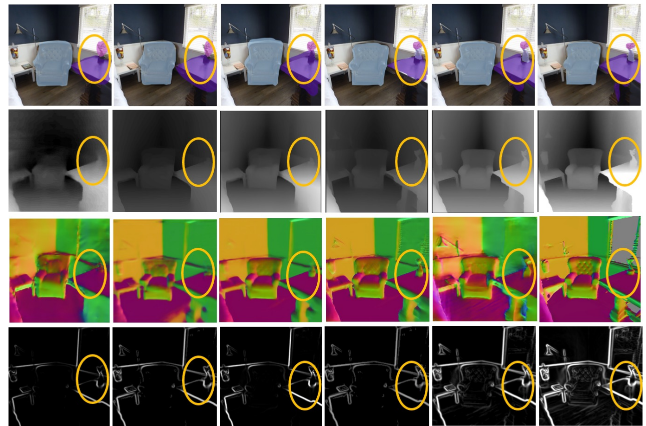

Figure 7: **Qualitative comparison** on Taskonomy benchmark [67] for *'S-D-N-E'*. From top to bottom, we show results on segmentation, depth, surface normal, and edges. Our model outperforms all the multitask baselines. We report the best-performing methods from Table 2. Best seen on screen and zoomed within the yellow circled regions.

4.2. Quantitative Results

Multitask Setting: Table 2 reports our main experimental results on two datasets where all models are initialized with the pre-trained ImageNet 22k model weights for a fair comparison. The baselines are selected based on their encoder-focused architectures that compare with our encoder-focused framework, as well as their task-affinity generalization, shown in Table 1. On Taskonomy [67] with the *'S-D-N-E'* labels, our method outperforms both the CNN-based [8, 29, 31, 34, 37, 52, 65, 67] and vision transformer-based MTL [35, 3] baselines by a considerable margin, showing the benefit of leveraging task-adapted attention. The same trend can be seen on Cityscapes [14]. Furthermore, we observe an increase in performance across all tasks with the addition of more tasks. This evidences the benefit of injecting additional geometrical cues in the form of surface normal or edges, respectively, to help the other tasks. We do not evaluate on IPT [7] because it was built to specifically solve deraining, denoising, and super-resolution. We also do not compare Vid-MTL [44] or UniT [21] as they cater to different modalities of learning such as video and text, respectively. The NYUDv2 [36], Synthia [42], Vkitti2 [4] MTL results are provided in the supplementary material.

	Methods	Taskonomy [67] 'S-D' SemSeg mIoU%↑	Taskonomy 'S-D' Depth RMSE↓	Taskonomy 'S-D-N' SemSeg mIoU%↑	Taskonomy 'S-D-N' Depth RMSE↓	Taskonomy 'S-D-N' Normal mErr.↓	Taskonomy 'S-D-N-E' SemSeg mIoU%↑	Taskonomy 'S-D-N-E' Depth RMSE↓	Taskonomy 'S-D-N-E' Normal mErr.↓	Taskonomy 'S-D-N-E' Edges F1%↑	Cityscapes [14] 'S-D' SemSeg mIoU%↑	Cityscapes 'S-D' Depth RMSE↓	Cityscapes 'S-D-N' SemSeg mIoU%↑	Cityscapes 'S-D-N' Depth RMSE↓	Cityscapes 'S-D-N' Normal mErr.↓
CNN	MTL-baseline [52]	41.22	0.5640	45.16	0.5398	29.30	47.64	0.5091	25.11	53.96	70.66	6.726	70.93	6.721	43.60
	Cross-stitch [34]	30.83	0.7780	32.84	0.7530	34.11	34.91	0.6980	32.04	37.58	50.33	7.683	54.99	7.311	44.10
	MTAN [29]	32.94	0.6800	35.79	0.6440	32.25	38.88	0.6030	30.55	40.19	53.86	7.318	57.23	7.050	42.09
	TTNet [37]	42.75	0.5270	45.85	0.5011	24.88	52.79	0.4872	24.11	56.10	71.00	6.655	71.40	6.511	41.23
	Taskonomy [67]	42.00	0.5633	43.11	0.5391	29.67	48.40	0.5086	26.62	53.99	60.02	7.204	63.41	7.044	41.91
	TSwitch [48]	42.79	0.5222	45.91	0.5007	24.82	52.81	0.4873	24.00	56.72	71.13	6.634	71.45	6.509	41.00
	Consistency [65]	42.46	0.5293	45.69	0.5013	27.22	52.55	0.4899	24.02	55.50	70.23	6.671	71.67	6.575	41.46
	XTAM [31]	43.24	0.4966	45.77	0.4888	25.05	52.71	0.4701	22.19	58.10	75.02	6.653	75.92	6.419	40.39
	TAWT [8]	44.07	<u>0.4935</u>	48.92	0.4833	24.86	53.15	0.4658	22.02	61.77	74.95	6.649	76.08	6.407	40.05
Transformer	ST-MTL [35]	45.12	0.4990	49.34	0.4750	23.11	53.17	0.4600	21.80	62.85	75.01	6.655	76.13	6.429	39.95
	MulT [3]	<u>49.73</u>	0.4981	<u>52.13</u>	<u>0.4501</u>	<u>21.86</u>	<u>54.04</u>	<u>0.4429</u>	<u>20.10</u>	<u>65.62</u>	<u>76.05</u>	6.650	<u>77.50</u>	<u>6.391</u>	<u>39.84</u>
	Our	**52.46**	**0.4524**	**57.03**	**0.4291**	**19.46**	**60.80**	**0.3903**	**17.13**	**71.09**	**78.00**	**6.503**	**80.55**	**6.307**	**39.05**

Table 2: **Quantitative comparison** on the Taskonomy [67] and Cityscapes [14] benchmarks for different multitask settings of *'S-D'*, *'S-D-N'* and *'S-D-N-E'*. Our model consistently outperforms both CNN-based and vision transformer-based MTL baselines. We show that adding more tasks improves their respective performances based on their task affinities. Bold and underlined values show the best and second-best results, respectively.

Zero-Shot Task Transfer: In Table 3, we apply the models trained on Vkitti2 [4] to Cityscapes [14]. As the name suggests, a model that infers a zero-shot task is not trained with *any* labels corresponding to that task. However, it should have a notion of the zero-shot task, which it leverages from the trained Vkitti2 labels. As shown in Table 3, our method outperforms all the baselines on zero-shot depth prediction and zero-shot normal prediction on Cityscapes by at least 0.196, and 1.59 points, respectively. For the zero-shot task transfer experiments, we compare with the baselines that have investigated task transfer learning [8, 37, 65, 67], shown in Table 1. We also compare with methods that use the same transformer backbone such as Vanilla MTL Swin [30] and MulT [3]. See the supplementary for the results on Synthia.

Although we have shown experiments on dense tasks throughout our paper, note that our model is not restricted to just dense tasks. In the supplementary, we further report our model's performance for the zero-shot image captioning task (IC) on the 'noCaps out-of-domain' benchmark.

Unsupervised Domain Adaptation: In this setting, the goal is to perform well on average on all tasks in the target domain, when the model is trained only on source domain labels but is *aware* of the target domain images. We argue that task adaptation is beneficial for multitasking UDA as semantic and geometrical tasks exhibit complementary behaviors. We report results for the typical synthetic to real scenario, namely Vkitti2→Cityscapes, in Table 4 for the *'S-D'* multitask setup. We adopt a simple multitask Domain Adaptation (DA) solution based on output-level DA adversarial training [43] for all the models. We also report the 1-task Swin-target (Oracle), trained on the labeled target data. The use of our vision transformer adapters' task-adaptation mechanism significantly improves performance on all metrics. The selected baselines for UDA evalua-

	Methods	'S-D' SemSeg mIoU%↑	'S-D' Depth RMSE↓	'S-D-N' SemSeg mIoU%↑	'S-D-N' Depth RMSE↓	'S-D-N' Normal mErr.↓
CNN	TTNet [37]	71.00	8.101	71.40	6.511	49.22
	Taskonomy [67]	60.02	8.694	63.41	7.044	53.57
	Consistency [65]	70.23	7.773	71.67	6.575	48.51
	TAWT [8]	75.02	<u>7.596</u>	75.92	6.419	45.28
Transformer	Vanilla MTL Swin [30]	75.10	8.003	75.97	8.000	49.05
	MulT [3]	<u>76.05</u>	7.115	<u>77.50</u>	<u>6.391</u>	<u>42.69</u>
	Our	**78.00**	**6.919**	**80.55**	**6.307**	**41.10**

Table 3: **Results on zero-shot task transfer**. Our method outperforms all the MTL baselines. All the methods are first trained on the Vkitti2 benchmark and then fine-tuned to Cityscapes [14]. Zero-shot task predictions are highlighted in blue and yellow, respectively. Bold and underlined values show the best and second-best results.

	Methods	MTL	SemSeg mIoU%↑	Depth RMSE↓
CNN	MTL-baseline-UDA [52]	✓	57.26	11.85
	Consistency-UDA [65]	✓	62.19	11.33
	XTAM-UDA [31]	✓	63.76	11.15
Transformer	1-task Swin-UDA [41]	✗	63.88	11.09
	MulT-UDA [3]	✓	<u>66.12</u>	<u>10.35</u>
	Our-UDA	✓	**70.93**	**08.66**
	1-task Swin-target (Oracle) [30]	✗	75.97	06.65

Table 4: **Unsupervised Domain Adaptation (UDA)** results for Vkitti2 [4]→Cityscapes [14]. Our model outperforms all the baselines. Bold and underlined values show the best and second-best results, respectively.

	Methods	MTL	SemSeg mIoU%↑	Depth RMSE↓
CNN	Consistency [65]	✓	26.24	0.771
	XTAM [31]	✓	29.13	0.750
Transformer	1-task Swin [30]	✗	32.09	0.722
	ST-MTL [35]	✓	32.51	0.720
	MulT [35]	✓	<u>33.68</u>	<u>0.701</u>
	Our	✓	**40.77**	**0.652**

Table 5: **Generalization to novel domains** results for Taskonomy [67]→NYUDv2 [36]. Our model outperforms all the baselines. Bold and underlined values show the best and second-best results, respectively.

tion are those that have investigated UDA in their respective works [3, 31, 52, 65]. Further details are provided in the supplementary material.

Generalization to Novel Domains: The TROA and TAA modules, in our vision transformer adapters, achieve generalization. In this section, we demonstrate how well our method generalizes to new domains without any fine-tuning. We compare our model with the two CNN-based MTL baselines of Consistency [65], XTAM [31], as well as the 1-task Swin baseline [30], and vision transformer-based MTL baselines such as ST-MTL [35], and MulT [3], reported in Table 5. We use the models trained on Taskonomy dataset [67] and apply them to the NYUDv2 [36] dataset without fine-tuning, as we find the task affinities are similar across these domains. For example, TROA finds how similar segmentation and depth tasks (c.f. Figure 4) are for Taskonomy comprising indoor scenes. This affinity when used together with TAA, ultimately, generalizes to NYU-v2 comprising indoor scenes. An intuitive observation is that none of these models generalize to extremely disparate domains i.e. the networks trained on indoor scenes from Taskonomy cannot generalize to datasets with 'faces' or 'animals', simply because the networks have no notion of such categories of data. Nonetheless, we study the generalizability of our method to a disparate comics domain when the network is trained on MS-Coco [28] which contains 'faces' or 'animals'. We provide these results in the supplementary.

	Model	# Parameters (Millions)↓	Training time (mins per epoch)↓
CNN	XTAM [31]	304	16
	Consistency [65]	<u>228</u>	<u>14</u>
Transformer	Vanilla MTL Swin [30]	348	18
	MulT [3]	447	22
	Our	**105.7**	**8**

Table 6: **Parameter and training time comparison** of our model on the Taskonomy [67] benchmark. Our method is more parameter efficient than all the MTL baselines.

Parameter Comparison In Table 6, we compare the time taken to train the models, in minutes per epoch. We show that our method is more parameter efficient than the CNN-based MTL baselines [31, 65], vanilla Swin model [30], and the transformer-based MTL approach [3] on *'S-D-N-E'*, thanks to the vision adapters' bottleneck network that decreases the computational requirement by over an order of magnitude. We defer ablations for different modules of our network, different network sizes, freezing of encoder layers, and placement of the adapters in the supplementary.

5. Conclusion and Limitations

Our method demonstrates the benefit of task-adaptive learning for generalizable multitasking. Across the four settings, our method outperforms not only CNN-based MTL methods but also vision transformer-based ones. This shows that our method mitigates task interference and negative task transfer while promoting more efficient parameter sharing. Driven by the generalizability of our model, we hope that our method can help to solve dense task predictions on domains with limited data labels such as comics. However, our framework has some limitations:

Data Dependency. Our model is data-intensive in the MTL setting. When trained on a limited amount of data, it may not achieve the same performance as reported in this work which is also the case for all the baselines. However, we generalize to other tasks and domains, unlike the baselines.

Unpaired Data. Our current MTL model is trained in a supervised manner, thereby needing paired data. Extending our methodology to an unsupervised paradigm for MTL is feasible, as in [50]. Besides addressing these limitations, employing different pre-training modalities, such as text or video as in [10], is also feasible.

Acknowledgement. This work was supported in part by the Swiss National Science Foundation via the Sinergia grant CRSII5−180359.

References

[1] Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless Fowlkes, Stefano Soatto, and Pietro Perona. Task2vec: Task embedding for meta-learning. *arXiv:1902.03545, cs.LG*, 2019. 2

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4, 6

[3] Deblina Bhattacharjee, Tong Zhang, Sabine Süsstrunk, and Mathieu Salzmann. Mult: An end-to-end multitask learning transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12031–12041, June 2022. 1, 2, 3, 5, 6, 7, 8, 9

[4] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv: 2001.10773*, 2020. 7, 8

[5] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. *CoRR*, abs/2103.14899, 2021. 1

[6] Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. Regionvit: Regional-to-local attention for vision transformers. *arXiv:2106.02689, cs.CV*, 2021. 3

[7] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. *arXiv: 2012.00364, cs.CV*, 2021. 1, 2, 7

[8] Shuxiao Chen, Koby Crammer, Hangfeng He, Dan Roth, and Weijie J Su. Weighted training for cross-task learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 2, 3, 4, 7, 8

[9] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine*

Learning (ICML), volume 80 of *Proceedings of Machine Learning Research*, pages 794–803, 2018. 2

[10] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022. 2, 9

[11] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 2378–2394, 2021. 1

[12] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *Proceedings of the Conference on Neural Neural Information Processing Systems (NeurIPS)*, 2021. 3

[13] Roberto Cipolla, Yarin Gal, and Alex Kendall. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7482–7491, 2018. 2

[14] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7, 8

[15] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12124–12134, June 2022. 1

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 1, 2, 3, 7

[17] Kshitij Dwivedi and Gemma Roig. Representation similarity analysis for efficient task taxonomy & transfer learning. *arXiv:1904.11740, cs.CV*, 2019. 2

[18] Christopher Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 2, 5

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv:1512.03385*, 2015. 3

[20] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. *arXiv preprint arXiv: 1902.00751*, 2019. 2

[21] Ronghang Hu and Amanpreet Singh. Unit: Multimodal multitask learning with a unified transformer. *arXiv: 2102.10772, cs.CV*, 2021. 1, 2, 7

[22] Aditya Jonnalagadda, William Yang Wang, B. S. Manjunath, and Miguel P. Eckstein. Foveater: Foveated transformer for image classification. *arXiv preprint arXiv: 2105.14173*, 2021. 1

[23] Jack Lanchantin, Tianlu Wang, Vicente Ordonez, and Yanjun Qi. General multi-label image classification with transformers. *arXiv preprint arXiv: 2011.14027*, 2020. 1

[24] Jan Eric Lenssen, Christian Osendorfer, and Jonathan Masci. Deep iterative surface normal estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11247–11256, 2020. 1

[25] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv: 2203.16527*, 2022. 2

[26] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4804–4814, 2022. 1

[27] Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv: 2111.11429*, 2021. 2

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the IEEE/CVF European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014. 7, 9

[29] Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. *CoRR*, abs/1803.10704, 2018. 1, 2, 3, 7, 8

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv: 2103.14030, cs.CV*, 2021. 1, 2, 3, 6, 8, 9

[31] Ivan Lopes, Tuan-Hung Vu, and Raoul de Charette. Cross-task attention mechanism for dense multi-task learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023. 1, 2, 3, 6, 7, 8, 9

[32] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *Openreview*, 2018. 5

[33] Vincent Michalski, Vikram Voleti, Samira Ebrahimi Kahou, Anthony Ortiz, Pascal Vincent, Chris Pal, and Doina Precup. An empirical study of batch normalization and group normalization in conditional computation. *arXiv preprint arXiv: 1908.00061*, 2019. 6

[34] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3994–4003, 2016. 1, 2, 3, 7, 8

[35] Eslam Mohamed and Ahmed El-Sallab. Spatio-temporal multi-task learning transformer for joint moving object detection and segmentation. *arXiv: 2106.11401, cs.CV*, 2021. 1, 2, 3, 7, 8, 9

[36] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the IEEE/CVF European Conference on Computer Vision (ECCV)*, 2012. 7, 8, 9

[37] Arghya Pal and Vineeth N Balasubramanian. Zero-shot task transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 3, 7, 8

[38] Arghya Pal and Vineeth N. Balasubramanian. Zero-shot task transfer. *CoRR*, abs/1903.01092, 2019. 2

[39] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. *arXiv preprint arXiv: 1709.07871*, 2017. 6

[40] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *arXiv: 2103.13413, cs.CV*, 2021. 1

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241. Springer, 2015. 8

[42] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016. 7

[43] Suman Saha, Anton Obukhov, Danda Pani Paudel, Menelaos Kanakis, Yuhua Chen, Stamatios Georgoulis, and Luc Van Gool. Learning to relate depth and semantics for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8197–8207, 2021. 8

[44] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Video multitask transformer network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshop (ICCVW)*, pages 1553–1561, 2019. 1, 2, 7

[45] Trevor Standley, Amir R. Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? *arXiv: 1905.07553, cs.CV*, 2019. 1, 2, 5

[46] Asa Cooper Stickland and Iain Murray. BERT and PALs: Projected attention layers for efficient adaptation in multi-task learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 5986–5995, 09–15 Jun 2019. 3

[47] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. *arXiv preprint arXiv:2105.05633*, 2021. 1

[48] Guolei Sun, Thomas Probst, Danda Pani Paudel, Nikola Popović, Menelaos Kanakis, Jagruti Patel, Dengxin Dai, and Luc Van Gool. Task switching network for multi-task learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8291–8300, 2021. 1, 2, 3, 8

[49] Yi Tay, Zhe Zhao, Dara Bahri, Donald Metzler, and Da-Cheng Juan. Hypergrid transformers: Towards a single model for multiple tasks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 3

[50] Ye Tian, Haolei Weng, and Yang Feng. Unsupervised multi-task and transfer learning on gaussian mixture models. *arXiv preprint arXiv: 2209.15224*, 2022. 9

[51] Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*, 2020. 5

[52] Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. Multi-task learning for dense prediction tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 1, 2, 3, 6, 7, 8, 9

[53] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. *Proceedings of the IEEE/CVF European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 30, 2017. 6

[55] Aria Wang, Michael Tarr, and Leila Wehbe. Neural taskonomy: Inferring the similarity of task-derived representations from brain activity. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. 2

[56] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *arXiv:2106.13797, cs.CV*, 2021. 2, 3, 7

[57] Wenxiao Wang, Lu Yao, Long Chen, Deng Cai, Xiaofei He, and Wei Liu. Crossformer: A versatile vision transformer based on cross-scale attention. *arXiv:2108.00154, cs.CV*, 2021. 1, 3

[58] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 2

[59] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[60] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. *arXiv:2104.06399, cs.CV*, 2021. 3

[61] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformer-based attention networks for continuous pixel-wise prediction. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision (ICCV), 2021. 1

[62] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformer-based attention networks for continuous pixel-wise prediction. *arXiv preprint arXiv: 2103.12091*, 2021. 1

[63] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv: 2107.00641, cs.CV*, 2021. 2, 3, 7

[64] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pages 5824–5836, 2020. 2

[65] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust learning through cross-task consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11197–11206, 2020. 1, 2, 3, 6, 7, 8, 9

[66] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3712–3722, 2018. 1, 2

[67] Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3, 7, 8, 9

[68] Wenqiang Zhang, Zilong Huang, Guozhong Luo, Tao Chen, Xinggang Wang, Wenyu Liu, Gang Yu, and Chunhua. Shen. Topformer: Token pyramid transformer for mobile semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1