Video Task Decathlon: Unifying Image and Video Tasks in Autonomous Driving

Thomas E. Huang[1] Yifan Liu[1] Luc Van Gool[1,2] Fisher Yu[1]
[1]ETH Zürich [2]KU Leuven

https://www.vis.xyz/pub/vtd

Abstract

Performing multiple heterogeneous visual tasks in dynamic scenes is a hallmark of human perception capability. Despite remarkable progress in image and video recognition via representation learning, current research still focuses on designing specialized networks for singular, homogeneous, or simple combination of tasks. We instead explore the construction of a unified model for major image and video recognition tasks in autonomous driving with diverse input and output structures. To enable such an investigation, we design a new challenge, Video Task Decathlon (VTD), which includes ten representative image and video tasks spanning classification, segmentation, localization, and association of objects and pixels. On VTD, we develop our unified network, VTDNet, that uses a single structure and a single set of weights for all ten tasks. VTDNet groups similar tasks and employs task interaction stages to exchange information within and between task groups. Given the impracticality of labeling all tasks on all frames and the performance degradation associated with joint training of many tasks, we design a Curriculum training, Pseudo-labeling, and Fine-tuning (CPF) scheme to successfully train VTDNet on all tasks and mitigate performance loss. Armed with CPF, VTDNet significantly outperforms its single-task counterparts on most tasks with only 20% overall computations. VTD is a promising new direction for exploring the unification of perception tasks in autonomous driving.

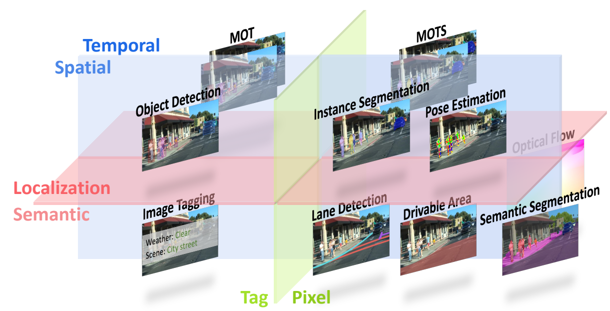

Figure 1: Task categorization of representative image and video recognition tasks. We design a new challenge and a new architecture to learn a unified representation of image and video tasks for autonomous driving.

1. Introduction

Agents that operate in dynamic environments are required to perform a wide range of visual tasks of varying complexities to carry out their functions. For instance, autonomous driving vehicles must identify drivable areas [49], detect pedestrians [32, 8], and track other vehicles [43, 55], among others. Taking a continuous stream of visual inputs, they must be capable of performing tasks at the level of images, instances, and instances across the spatial and temporal extent of the input data. While humans can effortlessly complete diverse visual tasks, and representation learning has shown impressive results on individual tasks [28], there is still a lack of unified architectures that can combine various heterogeneous tasks.

Unified representations for image and video tasks offer numerous advantages, including significant computational savings over using separate networks for each task [3]. Additionally, shared task input and output structures [53, 52] and cascaded tasks [35, 20] provide opportunities for learning algorithms to exploit inter-task relationships, resulting in better representations, generalization, and overall accuracy [38]. However, realizing these benefits poses unique challenges. Network architectures must support the predictions of all heterogeneous tasks, which is non-trivial due to the diversity in input and output structures and granularity of visual representation needed for each task. Furthermore, the impracticality of annotating all video frames for all tasks [55] results in data imbalance between each task and necessitates a more sophisticated training strategy than with single-task or homogeneous multi-task learning.

Another major obstacle to arrive at such a unified representation framework is the lack of large-scale evaluation

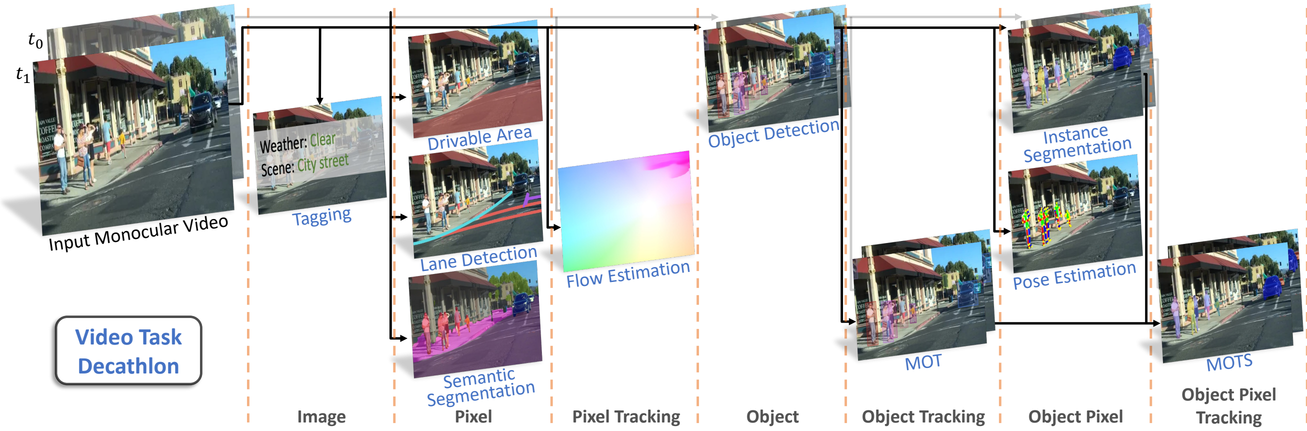

Figure 2: **Video Task Decathlon (VTD)**. We introduce the VTD to study unified representation learning of heterogeneous tasks in 2D vision for autonomous driving. Given a monocular video, the network needs to produce predictions for ten diverse image and video recognition tasks.

protocols for heterogeneous combinations of multiple tasks with distinct characteristics. Current multi-task benchmarks are overly simplistic and focus on combinations of multiple homogeneous tasks such as different types of classifications [36] or pixel-level predictions [34, 33, 7, 58]. Those that branch out to different tasks often only consider a limited number of tasks [2, 43, 13, 40, 25]. In addition, all these works are based solely on image tasks, disregarding the dynamics and associations in videos [57]. Although these benchmarks are useful for studying the abstract problem of multi-task learning, they do not adequately support the learning of general representations for the complex, real-world environments encountered in autonomous driving.

To address the aforementioned limitations, we first introduce a new challenge, **Video Task Decathlon (VTD)**, to study unified representation learning for heterogeneous tasks in autonomous driving. VTD comprises ten visual tasks, chosen to be representative of image and video recognition tasks (Figure 1). VTD provides an all-around test of *classification*, *segmentation*, *localization*, and *association* of objects and pixels. These tasks have diverse output structures and interdependencies, making it much more challenging than existing multi-task benchmarks. Additionally, differences in annotation density between tasks complicate optimization, reflecting real-world challenges. Along with our challenge, we also propose a new metric, VTD Accuracy (VTDA), that is robust to differing metric sensitivities and enables better analysis in the heterogeneous setting.

To explore unified representation learning on VTD, we propose two components: (1) **VTDNet**, a network capable of training on and producing outputs for every VTD task with a *single structure* and a *single set of weights*, and (2) **CPF**, a progressive learning scheme for joint learning on VTD. Specifically, VTDNet identifies three levels of visual features that are essential for visual tasks, namely image features, pixel features, and instance features. Each task can be broken down into a combination of these three basic features, and tasks are grouped based on the required features for prediction. Furthermore, VTDNet utilizes **Intra-group** and **Cross-group Interaction Blocks** to model feature interactions and promote feature sharing within and across different groups of tasks. CPF has three key features: **C**urriculum training pre-trains components of the network before joint optimization, **P**seudo-labels avoid forgetting tasks without sufficient annotations, and task-wise **F**ine-tuning boosts the task accuracies further based on the learned shared representations. CPF enables VTDNet to jointly learn all the tasks and mitigate a loss of performance.

We conduct experiments for the proposed VTD challenge on the large-scale autonomous driving dataset BDD100K [55]. Armed with CPF, VTDNet is able to significantly outperform strong baselines and other multi-task models on a majority of the tasks and achieve competitive performance on the rest, despite using a single set of weights and only 20% overall computations. Our findings indicate that unifying a diverse set of perception tasks for autonomous driving holds great promise for improving performance by leveraging shared knowledge and task relationships, while also achieving greater computational efficiency.

2. Related Work

Multi-Task Learning. Multi-task learning (MTL) [3] is the study of jointly learning and predicting several tasks. MTL may lead to performance gains due to knowledge sharing between different tasks and better generalization [38], while reducing the memory footprint and computational load. There are two main branches of research: architecture and optimization. With regard to architecture, early works studied joint learning of pairs of tasks, such as detection and segmentation [15]. Recent works focused on designing models that can learn shared representations from multiple

Table 1: Statistics of tasks and available annotations in BDD100K [55].

Set	Images (Train / Val)	% Total Images	Tasks with Annotations
Detection	70K / 10K	20%	Tagging, detection, pose, drivable area, lane detection
Segmentation	6.5K / 1K	2%	Instance segmentation, semantic segmentation
Tracking	280K / 40K (=1.4K / 200 videos)	78%	MOT, MOTS (partially annotated, 31K / 6.4K images)

different tasks [24, 51, 60, 47, 57, 25, 18], including Transformer [48] networks [53, 52, 59, 54]. Some study networks that can adaptively determine which parameters to share or use for tasks [31, 17], such as learning a layer-selection policy [45]. Others consider utilizing pseudo-labels [50, 12, 19]. In terms of optimization, most works focus on developing methods to automatically balance the losses of each task [41, 4, 21, 56, 27]. Some investigate better training strategies, such as task prioritization [14]. In this work, we extend the architectural study to the scale of ten heterogeneous image and video tasks for autonomous driving.

Multi-Task Benchmarks. Existing MTL datasets typically contain homogeneous image-based tasks and focus on either image classification [36] or dense prediction tasks [34, 33, 7, 58]. Recently, new datasets have been proposed to study the combination of object detection, monocular depth estimation, and panoptic segmentation [13, 40]. There are also large-scale autonomous driving datasets, mainly for studying detection and tracking [11, 2, 43]. Additionally, synthetic datasets have been developed for autonomous driving [39, 44]. Although these datasets provide a good foundation for the study of MTL, we argue they are either limited in scale or diversity of tasks. Conversely, BDD100K [55] is a large-scale autonomous driving dataset that contains labels for a diverse set of heterogeneous visual tasks that includes video tasks as well, which enables a new avenue for investigation. We design a new heterogeneous multi-task challenge based on BDD100K with ten diverse tasks to enable investigation of the unification of image and video tasks for autonomous driving.

3. Video Task Decathlon

We introduce Video Task Decathlon (VTD), a new challenge for investigating heterogeneous multi-task learning on a diverse set of 2D video tasks for autonomous driving (Figure 2). The goal is to facilitate designing models capable of handling all 2D tasks on monocular video frames. VTD comprises ten tasks: image tagging, object detection, pose estimation, drivable area segmentation, lane detection, semantic segmentation, instance segmentation, optical flow estimation, and multi-object tracking (MOT) and segmentation (MOTS). These tasks are representative of the space of 2D vision for autonomous driving (Figure 1).

Dataset. We build VTD on top of the real-world large-scale BDD100K video dataset [55], which has annotations for a diverse range of vision tasks. BDD100K consists of 100K driving video sequences, each around 40 seconds long. The tracking tasks are annotated at 5 FPS. The tasks are annotated on three separate image sets, which are all subsets of the original 100K videos. However, each image set only has labels for a portion of the available tasks. The statistics (after data deduplication) and tasks within each set are shown in Table 1. The varying size of each set reflects real-world difficulties in annotation. Consequently, this complicates optimization as different tasks have different data proportions and each image is only partially labeled.

3.1. Tasks

In this section, we describe each task in VTD. Due to space constraints, we omit task-specific details and include them in the supplementary material.

Image Tagging (G). Image tagging is composed of two classification tasks aiming to classify the input image into one of seven different weather conditions and seven scene types. We use the top-1 accuracy as the metric for each task ($\mathrm{Acc}^{\mathtt{Gw}}$ for weather and $\mathrm{Acc}^{\mathtt{Gs}}$ for scene).

Drivable Area Segmentation (A). The drivable area task involves predicting which areas of the image are drivable. We use the standard mIoU metric ($\mathrm{IoU}^{\mathtt{A}}$).

Lane Detection (L). The task of lane detection is to predict the lane types and their location in the image. We treat lane detection as a contour detection problem. For evaluation, we use the boundaries mIoU metric ($\mathrm{IoU}^{\mathtt{L}}$).

Semantic (S) / Instance Segmentation (I). Semantic and instance segmentation involves predicting the category and instance label for every pixel. We use the mIoU metric for semantic segmentation ($\mathrm{IoU}^{\mathtt{S}}$) and the mask AP metric for instance segmentation ($\mathrm{AP}^{\mathtt{I}}$).

Object Detection (D) / Pose Estimation (P). Object detection involves predicting a 2D bounding box and the category for each object in the image. Pose estimation involves localizing 2D joint positions for each human in the image. For evaluation, we use the bounding box AP metric ($\mathrm{AP}^{\mathtt{D}}$) for detection and Object Keypoint Similarity (OKS) AP metric ($\mathrm{AP}^{\mathtt{P}}$) for pose estimation.

MOT (T) / MOTS (R). MOT involves detecting and tracking every object in the video sequence. MOTS also includes segmentation of every object. For evaluation, we use a combination of AP for measuring detection and segmentation performance ($\mathrm{AP}^{\mathtt{T}}$ and $\mathrm{AP}^{\mathtt{R}}$) and AssA from HOTA [30] for measuring association performance ($\mathrm{AssA}^{\mathtt{T}}$ and $\mathrm{AssA}^{\mathtt{R}}$).

Optical Flow Estimation (O). Optical Flow estimation is the task of determining pixel-wise motions between pairs of

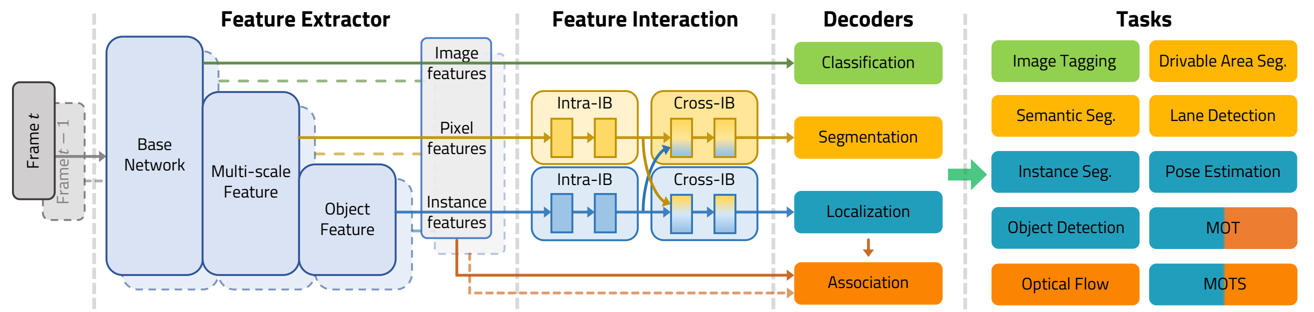

Figure 3: **Unified architecture of VTDNet.** Tasks are grouped into classification, segmentation, localization, and association. VTDNet includes a shared feature extractor to extract hierarchical features, feature interaction blocks to exchange knowledge between tasks, and decoders to make the final prediction for each task.

images. As BDD100K does not have labels for optical flow, we use a proxy evaluation method based on MOTS labels by warping the segmentation masks with the predicted flow and using the overlap with the ground-truth masks as the score [46]. We use mean IoU as the metric ($\mathrm{IoU}^{\mathsf{F}}$).

3.2. Evaluation

We here introduce our metric for evaluating model performances on VTD. There are two difficulties with designing a metric for a heterogeneous multi-task setup. First, as different tasks have different metrics, their sensitivity also varies, causing task-wise improvements to differ in scale. Second, due to the number of tasks and inter-class overlap, only looking at task-specific metrics will not give a clear indication of the model's overall performance, and a simple average over all metrics will hide task-wise differences.

To address these issues, we propose our VTD Accuracy (VTDA) metric. We first account for differing metric sensitivities by using the standard deviation in their measurements to scale their score accordingly. We estimate the standard deviation σ_t of each task t by measuring it over single-task baseline performances across different base networks (section 5.1), which informs how each metric's values change across increasing network capacity and differing architectures. Next, we discretize σ_t estimates to account for noise and convert them to a scaling factor by $s_t = 1/\lceil 2\sigma_t \rceil$, such that metrics with lower standard deviation will be scaled higher as differences are more significant and vice versa. This ensures that task scores contribute similarly to the final score and reduces bias towards a particular task.

Additionally, to better analyze multi-task performance, we separate the ten tasks into four groups first, each measuring a key aspect of the network's performance: classification, segmentation, localization, and association.

Classification. This group includes the two classification tasks in image tagging.

Segmentation. Segmentation refers to tasks that require prediction of a class label for each pixel in the image, *i.e.*, dense prediction. In VTD, this includes semantic segmentation, drivable area segmentation, and lane detection.

Localization. Localization includes object detection (bounding box), instance segmentation (pixel mask), and pose estimation (keypoints). We also consider detection and instance segmentation for object tracking (MOT and MOTS).

Association. Association includes optical flow estimation (image pixels), MOT (object bounding boxes), and MOTS (object pixel masks). In this group, we only evaluate the association performance of tracking, as localization errors are already accounted for in the localization group.

VTDA. We take the average of the scaled task performance within each group to compute a corresponding measure,

$$
\begin{aligned}
\mathrm{VTDA}_{\mathrm{cls}} &= (\mathrm{Acc}_s^{\mathsf{Gw}} + \mathrm{Acc}_s^{\mathsf{Gs}})/2, \\
\mathrm{VTDA}_{\mathrm{seg}} &= \left(\mathrm{IoU}_s^{\mathsf{S}} + \mathrm{IoU}_s^{\mathsf{A}} + \mathrm{IoU}_s^{\mathsf{L}}\right)/3, \\
\mathrm{VTDA}_{\mathrm{loc}} &= \left(\mathrm{AP}_s^{\mathsf{D}} + \mathrm{AP}_s^{\mathsf{I}} + \mathrm{AP}_s^{\mathsf{P}} + \mathrm{AP}_s^{\mathsf{T}} + \mathrm{AP}_s^{\mathsf{R}}\right)/5, \\
\mathrm{VTDA}_{\mathrm{ass}} &= \left(\mathrm{AssA}_s^{\mathsf{T}} + \mathrm{AssA}_s^{\mathsf{R}} + \mathrm{IoU}_s^{\mathsf{F}}\right)/3,
\end{aligned}
\tag{1}
$$

where the subscript s denotes scaling. Each score is normalized to the range [0, 100], and VTDA is defined as the sum of all scores. We provide full details about VTDA including the scaling factors used in the supplementary material.

4. Method

To tackle VTD, we propose VTDNet, a network capable of learning a unified representation for all ten tasks. We describe the network architecture in detail in section 4.1, feature interaction blocks in section 4.2, and the optimization protocol in section 4.3. The architecture is shown in Figure 3.

4.1. Heterogeneous Multi-Task Network

The heterogeneous nature of the VTD tasks necessitates input features to be extracted in a similar hierarchical manner, as different tasks require features at different visual granularities. VTDNet first uses a shared feature extractor to obtain three levels of visual features, namely image features, pixel features, and instance features. These features are essential for visual tasks and can be used to tackle the VTD tasks.

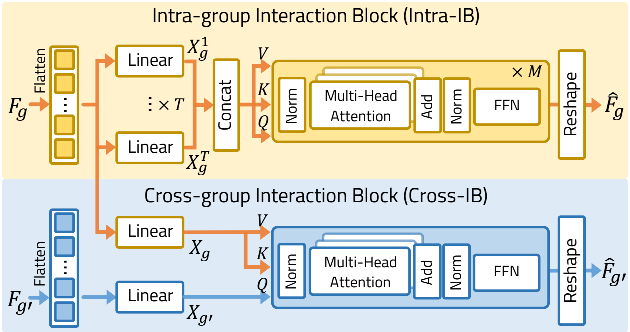

Figure 4: **Feature interaction blocks in VTDNet.** We use self- and cross-attention modules to model feature interactions within and between groups of tasks.

Additionally, we separate the tasks into four task groups following VTDA: classification, segmentation, localization, and association. Tasks in each group operate at the same feature level and can leverage shared processing to exchange information within the group (section 4.2). Independent decoders are used to produce the final predictions for each task. We detail the feature extractor and each decoder group below. To save space, we omit further details and elaborate them in the supplementary material.

Feature Extractor consists of the base network, a multi-scale feature extractor, and an object feature extractor to obtain hierarchical features. The base network produces image features $\{C2, C3, C4, C5\}$ with strides $\{2^2, 2^3, 2^4, 2^5\}$. Next, we use a Feature Pyramid Network (FPN) [26] to construct a multi-scale feature pyramid based on the image features and produce pixel features $\{P2, P3, P4, P5, P6\}$. Finally, we use a Region Proposal Network (RPN) [37] to produce instance features at each scale. For simplicity, we use an image-based base network rather than a video-based one, which enables VTDNet to operate online. For video tasks, we apply the same feature processing to additional video frames independently.

Classification Decoders operate on image features for prediction. The image tagging decoder uses global average pooling on $C5$ and has two dense layers, one for each classification task.

Segmentation Decoders require high resolution pixel features to output per-pixel predictions. We progressively upsample each FPN feature map to the same scale as $P2$ using convolutions and aggregate them with element-wise summation. Given the aggregated features, the drivable area, lane detection, and semantic segmentation decoders each use convolutional layers to obtain the final outputs.

Localization Decoders utilize instance features to make predictions for every object in the image. The detection, instance segmentation, and pose estimation decoders use parallel convolutional branches to map the instance features to the desired output [15].

Association Decoders are built on the previous features and decoder outputs across pairs of video frames. The flow estimation decoder uses warping on the features from the first two pixel feature maps $P2$ and $P3$ to construct a cost volume and convolutions to predict the flow, following standard procedure [42]. The MOT and MOTS decoders associate objects predicted by the detection and instance segmentation decoders using a learned similarity measure through contrastive learning, following QDTrack [10, 35].

Training Loss. Our joint training loss function is defined as a linear combination of all losses $L_{\texttt{VTD}} = \sum_t \lambda_t L_t$ for each task t with corresponding loss weight λ_t.

4.2. Feature Interaction

To further enhance knowledge sharing between tasks, we augment VTDNet with explicit pathways to incorporate additional avenues for task interactions and information exchange. We include such pathways by adding feature interaction blocks between similar tasks in the same group (intra-group) and between tasks in different groups (cross-group). These blocks are placed after the feature extractor and before the task decoders, and they are shown in Figure 4.

Intra-group Interaction Block (Intra-IB). Similar tasks within the same group can benefit from additional shared processing to model task interactions before independent task decoding. We incorporate interaction blocks based on attention [48] to model such interactions. Specifically, for a particular task group g and input features $F_g \in \mathbb{R}^{H\times W\times C}$, we first flatten into tokens and use a set of linear layers $L_g^1, \dots, L_g^T$ to extract task-specific tokens $X_g^1, \dots, X_g^T \in \mathbb{R}^{HW\times C'}$, where T is the number of tasks in group g. After concatenation $\hat{X}_g = \text{Concat}\left(X_g^1, \dots, X_g^T\right) \in \mathbb{R}^{HW\times TC'}$, we use a series of self-attention blocks for modeling interactions, each of which consists of Layer Normalization (LN) [1], Multi-Head Attention (MHA), and a feedforward network (FFN):

$$
\begin{aligned}
Q = \text{LN}(\hat{X}_g^i), K = \text{LN}(\hat{X}_g^i), V = \text{LN}(\hat{X}_g^i),\\
\hat{X}_g^{\prime i} = \text{MHA}(Q, K, V) + \hat{X}_g^i,\\
\hat{X}_g^{i+1} = \text{FFN}(\text{LN}(\hat{X}_g^{\prime i})) + \hat{X}_g^{\prime i},
\end{aligned}
\tag{2}
$$

where i indicates the i-th self-attention block and Q, K, V are the query, key, and value matrices. We use M such self-attention blocks ($M = 2$ in our experiments). Finally, we reshape $\hat{X}_g$ back to the input feature dimensions to obtain output features $\hat{F}_g \in \mathbb{R}^{H\times W\times TC'}$.

We use Intra-IB with the segmentation and localization decoder groups. Since, Intra-IB introduces more parameters and computation to the task group, we reduce the size of each decoder in the group to offset the increase in computation, which makes them more lightweight and enables VTDNet to maintain its advantage in efficiency.

Cross-group Interaction Block (Cross-IB). Tasks in different groups can also benefit from sharing feature representations. For example, instance features can inject more knowledge regarding foreground objects to the segmentation task group, while pixel features can provide more information regarding background regions for the localization task group. We integrate additional feature interaction blocks between different task groups. Specifically, for any two task groups g and g' with corresponding input features $F_g \in \mathbb{R}^{H\times W\times C}$ and $F_{g'} \in \mathbb{R}^{H'\times W'\times C'}$, we flatten and use a pair of linear layers L_g and $L_{g'}$ to obtain task group tokens $X_g \in \mathbb{R}^{HW\times C''}$ and $X_{g'} \in \mathbb{R}^{H'W'\times C''}$. For interaction, we use a cross-attention module to incorporate information from one task to another:

$$\begin{aligned} Q = \text{LN}(X_{g'}), K &= \text{LN}(X_g), V = \text{LN}(X_g), \\ \hat{X}'_{g'} &= \text{MHA}(Q, K, V) + X_{g'}, \\ \hat{X}'_g &= \text{FFN}(\text{LN}(\hat{X}'_{g'})) + \hat{X}'_{g'}, \end{aligned} \tag{3}$$

where the task tokens from one group is used to query the features of the other group. Finally, we reshape $\hat{X}'_g$ back to the input feature dimensions to obtain output features $\hat{F}'_g \in \mathbb{R}^{H\times W\times C}$. We place Cross-IB after Intra-IB to model interactions between the segmentation and localization task groups in both directions.

4.3. Joint Learning

There are two main optimization challenges in VTD: diversity in annotation density and in difficulty of tasks. Different tasks have different trade-offs between annotation variety and density. For example, detection is labeled on sampled frames from 100K videos, while tracking is labeled on only 2K videos, which has more frames in total but lower variety. Additionally, different tasks require different numbers of optimization steps to converge. These problems are further exacerbated by the large number of tasks. Naive joint optimization of all tasks will lead to significantly worse performance (section 5.1). To address these challenges, we construct a progressive training protocol called CPF, which has three key features: **C**urriculum training, **P**seudo-labeling, and **F**ine-tuning.

Curriculum Training. In order to handle the difference in difficulty of tasks, we follow a curriculum training protocol where we first pre-train VTDNet on a subset of the tasks then jointly train on all tasks. During pre-training, we train the localization and object tracking decoders on all relevant data, as they require more optimization steps. This enables the entire feature extractor and data-hungry task decoders (*e.g.*, MOT) to be initialized before joint training, which greatly improves final multi-task performance. After pre-training, we jointly train our model on all tasks. We use a set-level round-robin sampling scheme for data sampling, which samples a batch from each image set in order. At each step, only the weights of task decoders that receive corresponding data are updated. For efficiency, we do not oversample from any image set and cycle through each image only once per epoch.

Pseudo-labeling. Due to differences in annotation density between tasks, joint training with all ten tasks will lead to bias in performance towards the label-dominant tasks. The performance of tasks with a smaller proportion of labels (semantic segmentation and pose estimation in VTD) will thus suffer. To combat this issue, we utilize the single-task baselines to generate pseudo-labels to provide more labels for these tasks during joint training, which mitigates performance loss due to underfitting. We use the same training loss for pseudo-labels as the original task loss. As our goal is to address label deficiency, we only generate pseudo-labels for semantic segmentation and pose estimation.

Fine-tuning. During joint training, most task decoders will only receive a training signal periodically. This means the input feature distribution will have shifted before the next gradient update, making it difficult for each decoder to fully converge. Additionally, gradients from other tasks may also interfere with the training. To alleviate these issues, we further fine-tune each decoder on its task data after joint training while freezing the rest of the network (including shared blocks). This is akin to downstream fine-tuning with the learned shared representation and enables each decoder to obtain dedicated training without interference.

5. Experiments

We conduct extensive experiments on VTD to evaluate the effectiveness of VTDNet. We also provide ablation studies and visualizations.

Implementation Details. We use AdamW [22, 29] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay of 0.05 as the optimizer. We initialize all models with ImageNet pre-trained weights [9]. All models are trained for 12 epochs with a batch size of 16 and full crop size of 720×1280. We use a learning rate of 0.0001, decreasing by a factor of 10 at epochs 8 and 11. For augmentation, we use multi-scale training and flipping. We use the same data augmentations and learning rate for every task to reduce efforts needed for hyperparameter tuning.

5.1. Main Results

We compare multi-task performance and computational efficiency of VTDNet against single-task and multi-task baselines as well as other multi-task models on VTD.

Comparison to Baselines. For fair comparison, we first compare VTDNet to two baselines: single-task and multi-task. Single-task baselines use the same architecture described in section 4.1 for each task without the extra components used for other tasks and without interaction blocks. Each single-task baseline is trained solely on the corresponding task data without using other annotations, except for MOT/MOTS that uses detection/instance segmentation data

Table 2: Comparison of VTDNet against single-task (ST) and multi-task (MT) baselines on VTD validation set. CPF denotes our training protocol, and † denotes a separate model is trained for each task. VTDNet outperforms both ST and MT baselines on most tasks across both base networks and achieves significantly better VTDA. **Black** / *blue* indicate best / second best.

Method	Base Network	CPF	Classification Tagging Acc^{Gw}	Acc^{Gs}	$VTDA_{cls}$	Segmentation Sem. IoU^{S}	Driv. IoU^{A}	Lane IoU^{L}	$VTDA_{seg}$	Localization Det. AP^{D}	Ins. AP^{I}	Pose AP^{P}	MOT AP^{T}	MOTS AP^{R}	$VTDA_{loc}$	Association Flow IoU^{F}	MOT $AssA^{T}$	MOTS $AssA^{R}$	$VTDA_{ass}$	VTDA
ST Baselines†	ResNet-50		81.9	77.9	80.6	59.7	83.9	**28.4**	*56.7*	32.3	20.2	*37.0*	32.9	27.2	29.7	59.6	48.8	42.4	51.3	218.2
MT Baseline	ResNet-50	✗	**83.5**	79.2	**82.1**	45.1	*85.2*	26.7	54.1	*32.7*	*26.5*	29.6	31.2	28.5	30.0	**60.6**	46.9	41.6	50.7	216.9 (-1.3)
MT Baseline	ResNet-50	✓	83.0	*79.4*	81.8	*60.6*	*85.2*	25.9	56.4	32.5	26.4	35.0	*33.8*	*30.2*	*31.5*	*60.3*	*49.0*	*43.3*	*51.8*	*221.5 (+3.3)*
VTDNet	ResNet-50	✓	*83.2*	**79.7**	*82.0*	**63.8**	**85.4**	*27.8*	**57.8**	**33.4**	**27.1**	**39.7**	**34.7**	**31.6**	**32.9**	*60.3*	**50.1**	**45.1**	**52.7**	**225.3 (+7.1)**
ST Baselines†	Swin-T		82.8	78.9	81.5	60.0	83.9	26.0	55.8	34.4	22.6	**40.4**	33.5	28.4	31.4	57.5	50.0	42.8	51.0	219.7
MT Baseline	Swin-T	✗	**84.0**	79.8	**82.6**	45.9	85.4	**26.3**	54.2	33.3	26.8	33.3	30.5	28.5	30.4	57.5	47.8	41.0	49.7	217.0 (-2.7)
MT Baseline	Swin-T	✓	83.5	**80.0**	82.3	*61.7*	*85.6*	25.4	*56.5*	**35.5**	*27.8*	34.0	*35.3*	*31.4*	*33.0*	*58.0*	*50.4*	*44.9*	*51.9*	*223.7 (+4.0)*
VTDNet	Swin-T	✓	*83.8*	**80.0**	*82.5*	**64.5**	**85.9**	**26.3**	**57.5**	*35.4*	**28.5**	*40.2*	**35.8**	**32.2**	**34.1**	**60.3**	**50.5**	**45.1**	**52.8**	**226.9 (+7.2)**

Table 3: Comparison of VTDNet with multi-task models and single-task (ST) baselines using ResNet-50 on a subset of VTD tasks. † denotes a separate model is trained for each task.

Method	Tasks	Segmentation Sem. IoU^{S}	Driv. IoU^{A}	Lane IoU^{L}	Localization Det. AP^{D}	Ins. AP^{I}	Pose AP^{P}	MOT AP^{T}	MOTS AP^{R}	Association MOT $AssA^{T}$	MOTS $AssA^{R}$
ST Baselines†	ST	59.7	83.9	**28.4**	32.3	20.2	37.0	32.9	27.2	48.8	42.4
Semantic FPN [23]	S, A, L	59.2	83.9	24.9	–	–	–	–	–	–	–
Panoptic FPN [23]	S, I	58.5	–	–	–	19.7	–	–	–	–	–
MaskFormer [6]	S, I	55.9	–	–	–	10.4	–	–	–	–	–
Mask2Former [5]	S, I	62.8	–	–	–	19.9	–	–	–	–	–
Mask2Former [5]	S, A, L, I	59.7	84.8	**28.4**	–	17.3	–	–	–	–	–
Mask R-CNN [15]	D, P	–	–	–	32.7	–	35.2	–	–	–	–
Mask R-CNN [15]	D, I, P	–	–	–	31.2	24.6	33.1	–	–	–	–
QDTrack-MOTS [35]	D, I, T, R	–	–	–	32.1	23.1	–	32.9	27.2	48.8	42.4
VTDNet	VTD	**63.8**	**85.4**	27.8	**33.4**	**27.1**	**39.7**	**34.7**	**31.6**	**50.1**	**45.1**

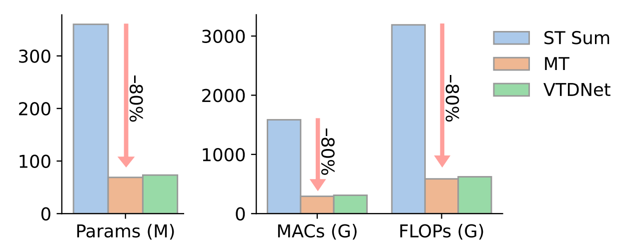

Figure 5: Comparison of resource usage during inference between VTDNet, single-task (ST), and multi-task (MT) baselines using ResNet-50 as the base network. Compared to ST baselines, VTDNet uses only one-fifth of the resources.

during training. We do not fix any other component besides the architecture and data, and we train each with task-specific augmentations and learning rate schedules, optimized for single-task performance. The multi-task baseline also uses the same architecture without interaction blocks, but it is optimized on all ten tasks jointly and uses all the VTD data. We provide complete details of every baseline in the supplementary material.

We conduct experiments on two different base networks: ResNet-50 [16] and Swin Transformer (Swin-T) [28]. The results are shown in Table 2. First, we find naive joint training with the multi-task baseline to not bring improvements to overall multi-task performance over the single-task baselines and severely hurts the accuracy of some tasks due to label deficiency (*e.g.*, pose estimation and semantic segmentation), task interference (*e.g.*, lane detection), and under-training (*e.g.*, MOT). This shows that a sophisticated training strategy is necessary to overcome optimization challenges in VTD. When optimizing the multi-task baseline with our CPF training protocol, significant performance gains can be achieved across the board, obtaining better scores than the single-task baselines on a majority of tasks and an improvement of over 3 points in VTDA. Furthermore, VTDNet is able to achieve additional improvements in performance over the baselines, obtaining the best scores on most tasks and an increase of over 7 points in VTDA across both base networks.

Comparison to Multi-Task Models. We compare VTDNet against various other MTL models trained on a subset of the VTD tasks in Table 3. We train these models with set-level batch sampling and task-specific data augmentations and schedules. While leveraging additional data from other tasks in a multi-task learning setting can boost per-task performance of a few tasks, performance on certain tasks (lane detection and pose estimation) greatly suffers due to task interference and label deficiency. In particular, we find Mask2Former [5] to perform well on semantic segmentation, but its instance segmentation performance is worse as it cannot take advantage of the abundant bounding box annotations – adding a detection decoder results in extremely poor detection accuracy of 17.8 AP^{D}. We further extend

Table 4: Ablation study of network components, including Inter-group (Intra-IB) and Cross-group (Cross-IB) Interaction Blocks with VTDNet using ResNet-50 on VTD validation set. All networks are trained with CPF.

Intra-IB	Cross-IB	Classification Tagging Acc^{Gw}	Acc^{Gs}	VTDA_{cls}	Segmentation Sem. IoU^{S}	Driv. IoU^{A}	Lane IoU^{L}	VTDA_{seg}	Localization Det. AP^{D}	Ins. AP^{I}	Pose AP^{P}	MOT AP^{T}	MOTS AP^{R}	VTDA_{loc}	Association Flow IoU^{F}	MOT AssA^{T}	MOTS AssA^{R}	VTDA_{ass}	VTDA
✗	✗	83.0	79.4	81.8	60.6	85.2	25.9	56.4	32.5	26.4	35.0	33.8	30.2	31.5	**60.3**	49.0	43.3	51.8	221.5
✓	✗	82.8	79.4	81.7	**63.9**	85.2	26.0	57.0	**33.4**	27.0	**39.8**	34.5	**31.7**	32.8	60.2	49.3	**45.1**	52.3	223.9 (+2.4)
✓	✓	**83.2**	**79.7**	**82.0**	63.8	**85.4**	**27.8**	**57.8**	**33.4**	**27.1**	39.7	**34.7**	31.6	**32.9**	**60.3**	**50.1**	**45.1**	**52.7**	**225.3 (+3.8)**

Table 5: Ablation study of CPF, including curriculum training (C), pseudo-labels (P), and fine-tuning (F) with VTDNet using ResNet-50 on VTD validation set. Highlighted improvements are underlined.

C	P	F	Classification Tagging Acc^{Gw}	Acc^{Gs}	VTDA_{cls}	Segmentation Sem. IoU^{S}	Driv. IoU^{A}	Lane IoU^{L}	VTDA_{seg}	Localization Det. AP^{D}	Ins. AP^{I}	Pose AP^{P}	MOT AP^{T}	MOTS AP^{R}	VTDA_{loc}	Association Flow IoU^{F}	MOT AssA^{T}	MOTS AssA^{R}	VTDA_{ass}	VTDA
✗	✗	✗	**83.6**	79.1	**82.1**	41.5	85.0	26.3	53.3	32.3	26.4	29.1	31.3	29.4	30.0	60.6	47.1	43.6	51.3	216.7
✓	✗	✗	83.0	79.4	81.8	42.0	84.9	26.2	53.3	**33.9**	**27.3**	31.1	31.9	30.0	31.1	**60.8**	47.7	43.8	51.6	217.8 (+1.1)
✓	✓	✗	83.2	79.6	82.0	63.2	85.0	26.1	56.8	33.7	27.1	39.0	31.7	30.4	31.9	60.1	48.0	44.5	51.7	222.4 (+5.7)
✓	✓	✓	83.2	**79.7**	82.0	**63.8**	**85.4**	**27.8**	**57.8**	33.4	27.1	**39.7**	**34.7**	**31.6**	**32.9**	60.3	**50.1**	**45.1**	**52.7**	**225.3 (+8.6)**

Table 6: Ablation study of different loss weight configurations with VTDNet using ResNet-50 on VTD validation set.

Loss Weights	VTDA_{cls}	VTDA_{seg}	VTDA_{loc}	VTDA_{ass}	VTDA
Default	82.0	57.8	32.9	52.7	**225.3**
$2\lambda_{\texttt{G}}$	**82.2**	57.6	32.4	52.5	224.7
$2\lambda_{\texttt{S}}, 2\lambda_{\texttt{A}}, 2\lambda_{\texttt{L}}$	81.9	**58.0**	32.5	52.3	224.7
$2\lambda_{\texttt{D}}, 2\lambda_{\texttt{I}}, 2\lambda_{\texttt{P}}$	82.1	57.3	**33.2**	52.5	225.0
$2\lambda_{\texttt{F}}, 2\lambda_{\texttt{T}}$	82.1	57.4	32.8	**53.0**	**225.3**

Mask2Former to handle other segmentation tasks and find it performs competitively, though performance on original tasks are degraded. In comparison, VTDNet can achieve further performance gains across all tasks while alleviating the performance loss, demonstrating the benefits of unifying the VTD tasks.

Resource Usage. We compare the resource usage during inference of VTDNet, single-task, and multi-task baselines with ResNet-50 base network in Figure 5. Since each single-task baseline uses a separate feature extractor, the computation accumulates with the increasing number of tasks. Comparatively, the multi-task baseline and VTDNet use around 80% fewer model parameters and operations due to sharing the feature extractor among all tasks. Additionally, since VTDNet replaces independent decoding layers in each task decoder with shared feature interaction blocks, VTDNet achieves significantly better performance with negligible computational overhead compared to the multi-task baseline.

5.2. Ablation Study and Analysis

We conduct a variety of ablation studies to evaluate different aspects of our network and our training protocol.

Network Components. We compare the effect of our feature interaction blocks, Intra-IB and Cross-IB, on VTDNet in Table 4. We train all networks with CPF for a fair comparison. Adding Intra-IB to model feature interactions between tasks in the same groups leads to significant improvements across segmentation and localization tasks, which results in an overall increase of 2.5 points in VTDA. In particular, performance on label-deficient tasks, semantic segmentation and pose estimation, is improved by over 3 points. Using Cross-IB further improves performance on segmentation tasks by 0.8 points on average, which leads to an additional 1.4 points increase in VTDA. This demonstrates that additional feature sharing within and between task groups can both largely benefit heterogeneous multi-task performance.

Training Protocol. We evaluate how components of our CPF protocol affect the final VTDNet performance on VTD in Table 5. Curriculum training significantly improves localization performance by pre-training the network before joint optimization. Using pose estimation and semantic segmentation pseudo-labels can completely resolve the label deficiency issue and result in much better performance on those tasks. Fine-tuning can further improve scores across the board by optimizing on task-specific data, especially for object tracking and segmentation tasks. By utilizing CPF, we can handle the optimization challenges of VTD and bring out the true benefits of multi-task learning.

Loss Weights and Metric. We investigate how VTDNet and VTDA behaves with various loss weight configurations in Table 6. Increasing or decreasing task loss weights results in the corresponding task group performance to increase or decrease, showing that one can modify the loss weights depending on the application to prioritize performance on certain tasks. VTDA can clearly demonstrate the improvements and decreases in performance of different aspects of the network. Furthermore, VTDA remains relatively stable across different configurations.

Visualizations. We show qualitative results of VTDNet on VTD validation set for several video sequences in Figure 6. The predictions of each task (excluding flow) are overlaid on top of each other in each frame. The color of each object indicate the predicted instance identity. For drivable area segmentation, red areas on the road indicate drivable regions, and blue areas indicate alternatively drivable areas. The green lines represent predicted lanes on the road. Across all

Figure 6: Visualization of VTDNet predictions on all tasks (excluding flow). Best viewed in color and zoomed in.

sequences, VTDNet can produce high-quality predictions that are consistent across all ten tasks with a single forward pass on each image.

6. Discussion and Conclusions

In this work, we present our new Video Task Decathlon (VTD) challenge to study heterogeneous multi-task learning for autonomous driving. VTD includes ten representative tasks on images and videos, allowing for the exploration of a unified representation for 2D vision. Our heterogeneous multi-task model VTDNet, equipped with feature interaction blocks and our CPF training protocol, significantly enhances the performance of single-task models while being much more efficient. We hope the VTD challenge can spark interest in this important area of research.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 2, 3

[3] Rich Caruana. Multitask learning. *Machine learning*, 1997. 1, 2

[4] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *ICML*, pages 794–803. PMLR, 2018. 3

[5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pages 1290–1299, 2022. 7

[6] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NeurIPS*, 34, 2021. 7

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 3

[8] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020. 1

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 6

[10] Tobias Fischer, Thomas E Huang, Jiangmiao Pang, Linlu Qiu, Haofeng Chen, Trevor Darrell, and Fisher Yu. Qdtrack: quasi-dense similarity learning for appearance-only multiple object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 5

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 3

[12] Golnaz Ghiasi, Barret Zoph, Ekin D Cubuk, Quoc V Le, and Tsung-Yi Lin. Multi-task self-training for learning general representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8856–8865, 2021. 3

[13] Kratarth Goel, Praveen Srinivasan, Sarah Tariq, and James Philbin. Quadronet: Multi-task learning for real-time semantic depth aware instance segmentation. In *WACV*, pages 315–324, January 2021. 2, 3

[14] Michelle Guo, Albert Haque, De-An Huang, Serena Yeung, and Li Fei-Fei. Dynamic task prioritization for multitask learning. In *ECCV*, September 2018. 3

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 2, 5, 7

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 7

[17] Menelaos Kanakis, David Bruggemann, Suman Saha, Stamatios Georgoulis, Anton Obukhov, and Luc Van Gool. Reparameterizing convolutions for incremental multi-task learning without task interference. In *ECCV*. Springer, 2020. 3

[18] Menelaos Kanakis, Thomas E Huang, David Bruggemann, Fisher Yu, and Luc Van Gool. Composite learning for robust and effective dense predictions. *arXiv preprint arXiv:2210.07239*, 2022. 3

[19] Menelaos Kanakis, Thomas E Huang, David Brüggemann, Fisher Yu, and Luc Van Gool. Composite learning for robust and effective dense predictions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2299–2308, 2023. 3

[20] Lei Ke, Xia Li, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Prototypical cross-attention networks for multiple object tracking and segmentation. *NeurIPS*, 34, 2021. 1

[21] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018. 3

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[23] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 7

[24] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, pages 6129–6138, 2017. 3

[25] Xiwen Liang, Yangxin Wu, Jianhua Han, Hang Xu, Chunjing Xu, and Xiaodan Liang. Effective adaptation in multi-task co-training for unified autonomous driving. *neurips*, 2022. 2, 3

[26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 5

[27] Liyang Liu, Yi Li, Zhanghui Kuang, Jing-Hao Xue, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Towards impartial multi-task learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 3

[28] 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. In *ICCV*, pages 10012–10022, 2021. 1, 7

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *iclr*, 2019. 6

[30] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *IJCV*, 129(2):548–578, 2021. 3

[31] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *CVPR*, pages 1851–1860, 2019. 3

[32] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 1

[33] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 2, 3

[34] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 2, 3

[35] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *CVPR*, 2021. 1, 5, 7

[36] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *NeurIPS*, 30, 2017. 2, 3

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28, 2015. 5

[38] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 1, 2

[39] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M Alvarez. Effective use of synthetic data for urban scene semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 84–100, 2018. 3

[40] Markus Schön, Michael Buchholz, and Klaus Dietmayer. Mgnet: Monocular geometric scene understanding for autonomous driving. In *ICCV*, 2021. 2, 3

[41] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *NeurIPS*, 31, 2018. 3

[42] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 5

[43] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov.

Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 3

[44] Tao Sun, Mattia Segu, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. SHIFT: a synthetic driving dataset for continuous multi-task domain adaptation. In *CVPR*, 2022. 3

[45] Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. Adashare: Learning what to share for efficient deep multi-task learning. *NeurIPS*, 2020. 3

[46] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J. Black. Video segmentation via object flow. In *CVPR*, pages 3899–3908, 2016. 4

[47] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *ECCV*, 2020. 3

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 3, 5

[49] Dong Wu, Manwen Liao, Weitian Zhang, and Xinggang Wang. Yolop: You only look once for panoptic driving perception. *arXiv preprint arXiv:2108.11250*, 2021. 1

[50] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020. 3

[51] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Padnet: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *CVPR*, June 2018. 3

[52] Xiaogang Xu, Hengshuang Zhao, Vibhav Vineet, Ser-Nam Lim, and Antonio Torralba. Mtformer: Multi-task learning via transformer and cross-task reasoning. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, page 304–321, Berlin, Heidelberg, 2022. Springer-Verlag. 1, 3

[53] Hanrong Ye and Dan Xu. Inverted pyramid multi-task transformer for dense scene understanding. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, pages 514–530. Springer, 2022. 1, 3

[54] Hanrong Ye and Dan Xu. Taskprompter: Spatial-channel multi-task prompting for dense scene understanding. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[55] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, pages 2636–2645, 2020. 1, 2, 3

[56] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *NeurIPS*, volume 33, pages 5824–5836. Curran Associates, Inc., 2020. 3

[57] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 2, 3

[58] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018. 2, 3

[59] Lefei Zhang et al. Demt: Deformable mixer transformer for multi-task learning of dense prediction. *arXiv preprint arXiv:2301.03461*, 2023. 3

[60] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Yan Yan, Nicu Sebe, and Jian Yang. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *CVPR*, pages 4106–4115, 2019. 3