Visual Exemplar Driven Task-Prompting for Unified Perception in Autonomous Driving

Xiwen Liang[1], Minzhe Niu[2], Jianhua Han[2], Hang Xu[2], Chunjing Xu[2], Xiaodan Liang[1†]
[1]Shenzhen Campus of Sun Yat-sen University, [2]Huawei Noah's Ark Lab
{liangxw29@mail2, liangxd9@mail}.sysu.edu.cn,
{niuminzhe1, hanjianhua4, xu.hang, xuchunjing}@huawei.com

Abstract

Multi-task learning has emerged as a powerful paradigm to solve a range of tasks simultaneously with good efficiency in both computation resources and inference time. However, these algorithms are designed for different tasks mostly not within the scope of autonomous driving, thus making it hard to compare multi-task methods in autonomous driving. Aiming to enable the comprehensive evaluation of present multi-task learning methods in autonomous driving, we extensively investigate the performance of popular multi-task methods on the large-scale driving dataset, which covers four common perception tasks, i.e., object detection, semantic segmentation, drivable area segmentation, and lane detection. We provide an in-depth analysis of current multi-task learning methods under different common settings and find out that the existing methods make progress but there is still a large performance gap compared with single-task baselines. To alleviate this dilemma in autonomous driving, we present an effective multi-task framework, VE-Prompt, which introduces visual exemplars via task-specific prompting to guide the model toward learning high-quality task-specific representations. Specifically, we generate visual exemplars based on bounding boxes and color-based markers, which provide accurate visual appearances of target categories and further mitigate the performance gap. Furthermore, we bridge transformer-based encoders and convolutional layers for efficient and accurate unified perception in autonomous driving. Comprehensive experimental results on the diverse self-driving dataset BDD100K show that the VE-Prompt improves the multi-task baseline and further surpasses single-task models.

1. Introduction

Multi-task learning (MTL) has been the source of a number of breakthroughs in autonomous driving over the last few years [23, 50, 56] and general vision tasks recently [2, 12, 26, 34, 55]. As the foundation of autonomous driving, a robust vision perception system is required to provide critical information, including the position of traffic participants, traffic signals like lights, signs, lanes, and obstacles that influence the drivable space, to ensure driving safety and comfort. These tasks gain knowledge from the same data source and present prominent relationships between each other, like traffic participants, are more likely to appear within drivable spaces and traffic signs may appear near traffic lights, etc. Training these tasks independently is time costing and fails to mine the latent relationship among them. Therefore, it is crucial to solve these multiple tasks simultaneously, which can improve data efficiency and reduce training and inference time.

Some recent works have attempted to apply unified training on multiple tasks in autonomous training. Uncertainty [21] trains per-pixel depth prediction, semantic segmenta-

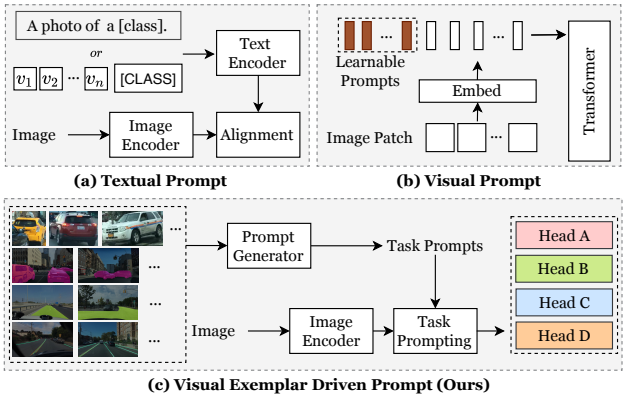

Figure 1. **Comparison of different prompts in computer vision.** (a) Extracting textual prompts from a text encoder to perform image-text alignment [62]. (b) Prepend learnable prompts to the embeddings of image patches [20]. (c) Visual exemplar driven prompts for multi-task learning (ours). The generated task prompts encode high-quality task-specific knowledge for downstream tasks.

†Corresponding author.

tion, and instance segmentation in a single model. CIL [19] introduces an extra traffic light classifier to learn different traffic patterns following traffic light changes. CP-MTL [4] learns object detection and depth prediction together to identify dangerous traffic scenes. However, these works differ in task types, evaluation matrix, and dataset, making it hard to compare their performances. For example, most of them are developed upon dense prediction [2, 55] and natural language understanding [7,47], rather than being tailored for more common perception tasks for autonomous driving, thus these methods may produce poor results when applied to a self-driving system. As a result, there is an emerging demand for a thorough evaluation of existing multi-task learning methods covering common tasks in autonomous driving.

In this paper, we focus on heterogeneous multi-task learning in common scenarios of autonomous driving and cover popular self-driving tasks, i.e., object detection, semantic segmentation, drivable area segmentation, and lane detection. We provide a systematic study of present MTL methods on large-scale driving dataset BDD100K [58]. Specifically, we find that task scheduling [26] is better than zeroing loss [51], but worse than pseudo labeling [15] on most tasks. Interestingly, in task-balancing methods, Uncertainty [21] produces satisfactory results on most tasks, while MGDA [41] only performs well on lane detection. This indicates that negative transfer [8], which is a phenomenon that increasing the performance of a model on one task will hurt the performance on another task with different needs, is common among these approaches.

To mitigate the negative transfer problem, we introduce the visual exemplar-driven task-prompting (shorten as **VE-Prompt**) based on the following motivations: (1) Given the visual clues of each task, the model can extract task-related information from the pre-trained model. Different from current prompting methods which introduce textual prompts [6, 38, 62, 63] or learnable context [20], we leverage exemplars containing information of target objects to generate task-specific prompts by considering that the visual clues should represent the specific task to some extent, and give hints for learning task-specific information; (2) Transformer has achieved competitive performance on many vision tasks but usually requires long training time, thus tackling four tasks simultaneously on a pure transformer is resource-intensive. To overcome this challenge, we efficiently bridge transformer encoders and convolutional layers to build the hybrid multi-task architecture. Extensive experiments show that VE-Prompt surpasses multi-task baselines by a large margin.

We summarize the main contributions of our work below:

- We provide an in-depth analysis of current multi-task learning approaches under multiple settings that comply with real-world scenarios, consisting of three common multi-task data split settings, two partial-label learning approaches, three task scheduling techniques, and three task balancing strategies.
- We propose an effective framework VE-Prompt, which utilizes visual exemplars to provide task-specific visual clues and guide the model toward learning high-quality task-specific representations.
- The VE-Prompt framework is constructed in a computationally efficient way and outperforms competitive multi-task methods on all tasks.

2. Related Work

Multi-task Learning Multi-task learning jointly trains shared parameters on multiple tasks, mining latent information among them to improve efficiency and accuracy. Famous multi-task learning models include Mask R-CNN [16], which applies Faster R-CNN [39] as the backbone and conducts instance segmentation and object detection at the same time. Other methods like Eigen *et al.* [11] address depth prediction, surface normal estimation, semantic labeling tasks, and MultiNet [44] provide prediction on classification, detection, semantic segmentation tasks within a single model. YOLOP [50] leverages CSPDarknet as the backbone, which branches out three task-specific heads for object detection, drivable area segmentation, and lane detection prediction. Standley *et al.* [42] and Christopher *et al.* [12] improves previous multi-task training schema by grouping proper tasks together rather than naively training all tasks together. In this paper, we focus on developing general and effective approaches for multi-task learning in autonomous driving scenarios.

Visual Perception for Autonomous Driving Autonomous driving relies on a perception system to gather information and understand the environment. Visual perception, as the most similar sensing modality to humans, provides high-resolution images that satisfy almost all tasks required for autonomous driving. Some of the tasks have long been studied beyond autonomous driving scenarios. Chen *et al.* [3] predicts 2D object detection from images while Semantic FPN [22] performs semantic segmentation and Lanenet [48] implements lane detection respectively using visual inputs. Though these models are designed for different tasks, they all adopt the backbone-header architecture, some of which even share the same backbone structure like ResNet [17] or transformer [10]. Running independent models for perception tasks separately is a waste of time and computation resources, making an emerging call for the development of a unified perception system.

Prompt-based Learning Prompt-based learning [18,30, 49] is put forward to bridge the gap between pre-training and model tuning in the field of natural language processing. GPT-3 [1] first designs various text prompts according

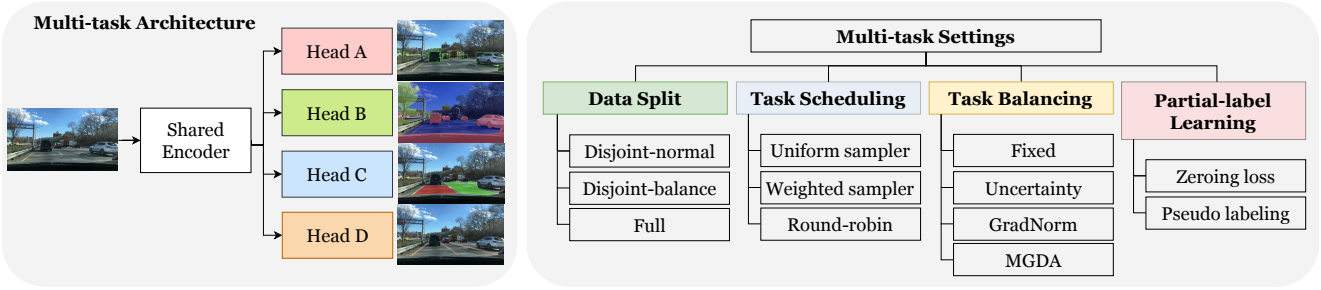

Figure 2. **The multi-task architecture and settings in our investigation.** We follow the common multi-task architecture where each task shares the same encoder and has its specific head. The multi-task settings focus on three types of task scheduling, four task balancing methods, and two partial-label learning techniques and cover three common data split settings.

to the property of tasks and treats the downstream task as a masked language modeling problem. Meanwhile, other approaches like [24,32,61] train learnable continuous prompts in the embedding space of the model and achieve competitive performance compared with finetuning. Recently, CLIP [37], which is trained on multi-modality vision-language pairs data, achieved impressive performance on zero-shot image classification by injecting visual categories into the text input as prompts. Subsequent works [13,57,62] further tune CLIP with learnable soft prompts by few-shot supervisions in the field of computer vision, or leverage text features from CLIP to enhance visual representations [6, 38]. Prompt tuning without textual information is introduced by injecting learnable vectors in the input space [20] or inserting lightweight blocks to learn prompts [36]. However, these approaches are tailored for solving downstream tasks independently and are inapplicable to heterogeneous multi-task learning. In this work, we design the visual exemplar-driven task-prompting to inject task-specific knowledge for heterogeneous multi-task learning.

3. Empirical Study

Multi-task Architectures Multi-task learning (MTL) architectures apply parameter sharing to learn shared information between different tasks. MTL architectures can be divided into encoder-focused architectures [14, 31, 35, 40] and decoder-focused ones [45, 54, 60] according to parameter sharing scope. Encoder-focused architectures can be further categorized into hard and soft parameter sharing. In this paper, we select the hard parameter-sharing structure as our backbone due to its simplicity and stability. Parameters are only shared in the encoder part of the model followed by task-specific heads. As Figure 2 shows, the image inputs first go through the shared encoder, and then the feature map is fed into different heads to produce corresponding predictions.

Task Scheduling Task scheduling is the process of choosing which task or tasks to train on at each training step. Some scheduling methods arrange the task orders during the training process in a fixed order like Round-Robin [58], while others may sample tasks following specific distributions [27], like Uniform sampler and Weighted sampler. Specifically, Uniform sampler samples tasks from a uniform distribution and Weighted sampler samples tasks with weight proportional to the number of training epochs of each task. We test the above three task scheduling methods in our investigation and compare their performances.

Task Balancing Task balancing is designed to deal with the gradients between tasks for the shared parameters in the network. When dealing with multiple tasks, the shared parameters are likely to be dominated by the one with a large gradient magnitude or confused by conflict gradients. It is intuitive to apply weights over these gradients to balance among tasks, and several methods have been proposed, including 1) Fixed weighting, which fixes all loss weights during training; 2) Uncertainty weighting [21], introducing the task-dependent Homoscedastic uncertainty as the basis for weighting losses by maximizing the Gaussian likelihood of the uncertainty; 3) GradNorm [5], calculating the product of $L2$ norm of task gradient and the relative inverse learning rate as the indicator of the task learning pace, and then setting task weights to minimize the learning pace difference among tasks to balance the training process; 4) MGDA [41], treating the Multi-Task Learning problem as a multi-objective optimization problem by using multiple gradient descent algorithm [9]; 5) ParetoMTL [29], which finds a solution called Pareto optimal solution where all task losses can decrease without increasing the loss on other tasks.

Learning on Partial Labels Image segmentation task requires annotations of labels to every pixel of the image, which costs a great time, and as a result hard to get enough annotations. To process the missing annotation problem, two different methods are introduced, including Zeroing loss [23,52] and Pseudo labeling [15]. Zeroing loss [23,52] simply zero losses for a particular task if the input image does not have the corresponding annotation. Pseudo labeling [15] first trains a teacher model on fully labeled data. Then the teacher model is used to label the missing annotations to create a multitask pseudo-labeled dataset.

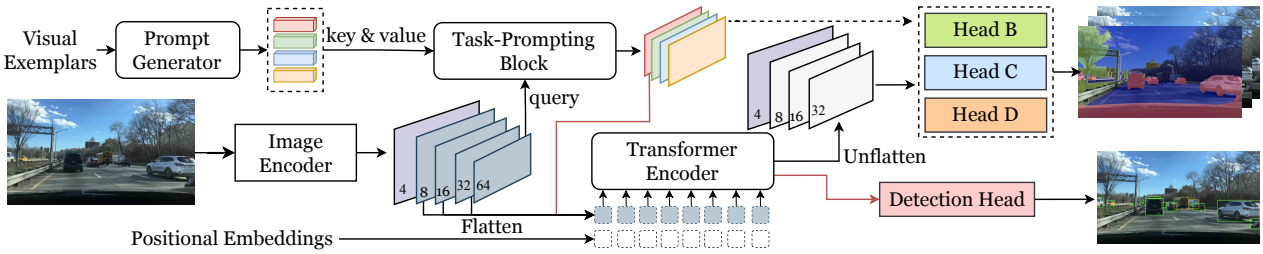

Figure 3. **The architecture of the proposed VE-Prompt.** VE-Prompt consists of (1) the image encoder to extract image features; (2) a shared transformer encoder for feature enhancement; (3) task-specific prompts generated by the prompt generator with visual exemplars; (4) a task-prompting block to integrate the visual representation with task-specific prompts; and (5) task-specific heads for different tasks.

We focus on four major tasks in autonomous driving, i.e., object detection, semantic segmentation, drivable area segmentation, and lane detection. The in-depth analysis of current multi-task methods is shown in Section 5.3.

4. VE-Prompt

The key to multi-task learning is to learn high-quality task-specific representations among tasks, which can explore relationships between tasks. Therefore, a good multi-task learning framework should take full advantage of task-specific priors, and guide the model to learn better representations. To this end, we introduce our proposed multi-task framework with VE-Prompt, which consists of five components: (1) an image feature encoder to extract image features; (2) a lightweight shared transformer encoder for feature enhancement; (3) task-specific prompts which encodes task-specific information from visual exemplars; (4) a visual exemplar driven task-prompting block to integrate the visual representation with task-specific prompts; (5) task-specific heads for predicting results simultaneously.

4.1. Bridging CNN and Transformer

The multi-task framework aims to learn more effective representations for all tasks via bridging CNN and Transformer efficiently. The neck of the image encoder and segmentation heads of the framework are CNN-based, reducing the overall training time. The shared transformer encoder is built upon the transformer architecture to capture the long-range dependency [53].

Image Encoder The image encoder consists of a backbone network and a neck network. We choose the Swin transformer [33] as the backbone to extract features of the input image. The output of the backbone is denoted as $\{C_2, C_3, C_4, C_5\}$. Then we adopt Feature Pyramid Network (FPN) [28] module for the neck network to fuse features generated by the backbone. The pyramidal features are of 5 scales, and the detection head only processes the last four-scale features for reducing the computation cost. Here we denote the output of the neck as $\{P_2, P_3, P_4, P_5, P_6\}$, which have strides of {4, 8, 16, 32, 64} pixels.

Shared Transformer Encoder The shared transformer encoder TransEncoder receives multi-scale outputs from the neck and enhances features for following task-specific heads. We first flatten the feature maps from $\{P_3, P_4, P_5, P_6\}$ and concatenate them to obtain a 1D sequence P. Since flattening the features leads to losing the spatial information critical for segmentation, we supplement positional embeddings p_l to the flattened features. For the model not considering prompts, we obtain the enhanced feature as follows:

$$O = \text{TransEncoder}(P + p_l). \tag{1}$$

After feature enhancement, O is passed to the detection head directly, while unflattened to multi-scale features $\{z_3, z_4, z_5, z_6\}$ for segmentation heads.

Detection Head The detection head consists of 4 multi-scale deformable decoder layers which are elaborated in DINO [59]. Following DINO, we adopt the mixed query selection strategy to initialize anchors as positional queries for the decoder and use the contrastive denoising training approach by taking into account hard negative samples.

Segmentation Head For segmentation-based tasks, we choose Semantic FPN [22] as the segmentation head. In the model without considering prompts, segmentation heads take in multi-layer features from both the neck and shared transformer encoder $\{P_2, z_3, z_4, z_5\}$. The resolution of P_2 is larger and thus provides more image information for the following heads. Then the multi-layer features are upsampled and summed element-wisely. This merged feature map is again upsampled $4\times$ followed by softmax to produce the classification score for every pixel at the original resolution.

4.2. Prompt Generation with Visual Exemplar

In order to motivate the model to learn more high-quality task-specific knowledge and handle all tasks better, VE-Prompt is introduced to provide more task-specific information with visual clues. The process of generating visual exemplar-driven prompts is shown in Figure 4. The key idea of task-specific prompts is to let the model know how to

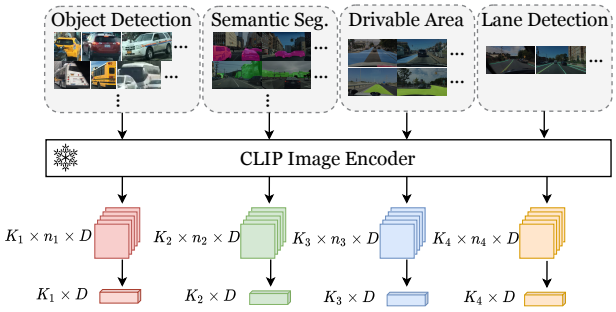

Figure 4. **Process of generating visual exemplar-driven prompts.** For the box-wise task, we crop class-related image regions to generate visual exemplars. For pixel-wise tasks, we mask class-related image regions with colored segmentation masks to produce exemplars. Then the fixed CLIP image encoder is adopted to extract task-specific prompts.

solve different tasks and what categories to focus on in advance of each task. Therefore, task-specific prompts should contain object-level information which helps the model understand tasks better, and we leverage visual exemplars to generate prompts.

We first sample a few examples from the training set to generate object-level image regions and segmentation masks as in Figure 4. There are a few generated exemplars for each category in each task, thus the generated prompts cover all classes. Then we adopt CLIP [37] to generate task-specific prompts since it is a robust feature extractor pre-trained with a huge amount of image-text data pairs. For visual perception, the ground-truth annotations can provide hints of the shapes and sizes of different objects, motivating the model to learn high-quality task-specific representations. The task-specific prompt is object-level for object detection and aims to represent relevant image regions. Only the generated prompts are used during training and inference, and no new exemplars will be included, thus it will not lead to training data leakage.

For the box-wise task (object detection), we use the annotated bounding boxes to crop sampled images and obtain raw object-level image regions for generating prompts. We choose the image encoder with ViT [10] backbone and pass n image regions $\{r_i^k\}$ of K classes to get the initial prompt as follows:

$$\begin{aligned} \{\hat{p_i^k}\} &= \text{L2\_NORM}(\text{IE}(\{r_i^k\})) \in \mathbb{R}^{K\times n\times D}, \\ p &= \frac{1}{n}\sum_{i}^{n}\{\hat{p_i^k}\} \in \mathbb{R}^{K\times D}, i = 1, 2, ..., n, \end{aligned} \tag{2}$$

where IE and D represent the image encoder of CLIP and the feature dimension respectively. n stands for the number of visual exemplars for each category. $\{\hat{p_i^k}\}$ and p indicate all prompts from image regions and the averaged version respectively. Specifically, class numbers for detection, semantic segmentation, drivable area segmentation, and lane detection are denoted as K_1, K_2, K_3, and K_4. The number of visual exemplars for different tasks is further denoted as n_1, n_2, n_3, and n_4.

For pixel-wise tasks (i.e., semantic segmentation, drivable area segmentation, and lane detection), image regions of specific classes are marked with colored segmentation masks, and different colors indicate different object categories of different tasks. Similar to object detection, after obtaining n images with colored segmentation masks of K classes, we adopt CLIP to extract features by Equation 2. In this way, we get task-specific prompts p_{det}, p_{sem}, p_{driv}, p_{lane} for object detection, semantic segmentation, drivable area segmentation, and lane detection, respectively.

4.3. Visual Exemplar Driven Task Prompting

Task prompting aims to integrate the image features with task-specific prompts to obtain high-quality task-specific representations. It receives additional task-specific prompts as inputs and generates task-specific features for following task-specific heads.

Here we design two prompting methods to improve task-specific representations. The first strategy is pre-head prompting. The last feature map P_6 from the neck and task-specific prompt p are fused via a transformer decoder:

$$f_{pre} = \text{TransDecoder}(q = P_6, k = p, v = p), \tag{3}$$

where q, k and v stand for query, key and value. In this way, we get task-specific features f_{pre}^{det}, f_{pre}^{sem}, f_{pre}^{driv}, and f_{pre}^{lane} for object detection, semantic segmentation, drivable area segmentation, and lane detection, respectively. For object detection, we flatten $\{P_3, P_4, P_5, f_{pre}^{det}\}$ to a 1D sequence and combine it with positional embeddings. Following Equation 1, we obtain features for the detection head O_{det}. For segmentation-based tasks, we first flatten $\{P_3, P_4, P_5\}$ to a 1D sequence and get the enhanced features through Equation 1. Then unflatten the output features as $\{z_3', z_4', z_5'\}$, and pass $\{P_2, z_3', z_4', z_5', f_{pre}]$ to specific segmentation-based heads. Note that f_{pre} is marked as f_{pre}^{sem}, f_{pre}^{driv}, or f_{pre}^{lane} according to the task type.

Another choice is to refine predicted results with task-specific prompts, namely post-head prompting. In this variant, we obtain class-related features ($\mathbb{R}^{K\times D}$) similar as follows:

$$f_{post} = \text{TransDecoder}(q = p, k = P_6, v = P_6). \tag{4}$$

Here class-related outputs from task-specific heads are denoted as v. Then the final output is calculated as:

$$v' = \text{MLP}(v \cdot f_{post}). \tag{5}$$

Empirical results of these two strategies are presented in Section 5.5 and show that pre-head prompting performs better than post-head prompting.

Table 1. Comparisons of popular task scheduling strategies and partial-label learning methods.

Setting	Methods	mAP	AP50	AP75	mIoU (SS)	mIoU (DA)	IoU (LD)	Avg.	$\Delta_{MTL}(\%)$
Full	Zeroing loss [51]	36.2	61.6	35.9	58.6	89.3	23.8	52.0	-2.68
	Pseudo labeling [15]	36.3	61.6	36.1	60.9	89.3	23.8	52.6	-1.65
	VE-Prompt (Ours)	**39.2**	**64.9**	**39.0**	**63.2**	**89.4**	**24.0**	**54.0**	**+1.52**
Disjoint-normal	Zeroing loss [51]	31.1	54.3	30.2	55.7	88.0	22.2	49.3	-2.64
	Uniform sampler [26]	30.1	52.8	29.0	60.6	88.6	23.4	50.7	-0.10
	Weighted sampler [26]	29.3	51.9	28.7	58.5	**88.9**	**23.8**	50.1	-1.19
	Round-robin [26]	30.2	53.1	29.7	61.0	88.7	23.5	50.9	+2.87
	Pseudo labeling [15]	32.6	54.6	32.3	59.7	88.2	23.0	50.9	+1.19
	VE-Prompt (Ours)	**34.2**	**56.9**	**33.9**	**62.2**	88.3	23.3	**52.0**	**+3.95**
Disjoint-balance	Zeroing loss [51]	29.7	52.3	29.2	57.5	86.7	21.4	48.8	-1.61
	Uniform sampler [26]	28.1	50.2	27.5	60.4	87.1	**22.6**	50.0	-0.44
	Round-robin [26]	28.4	50.8	27.8	60.0	87.1	**22.6**	49.5	-0.34
	Pseudo labeling [15]	31.3	52.8	30.8	60.2	87.0	22.2	50.2	+1.87
	VE-Prompt (Ours)	**33.9**	**56.6**	**33.7**	**61.2**	**87.4**	22.2	**51.2**	**+4.72**

Table 2. Comparisons of task balancing strategies with pseudo labels. * means using the full image encoder to compute the gradient norm.

Setting	Method	mAP	AP50	AP75	mIoU (SS)	mIoU (DA)	IoU (LD)	Avg.	$\Delta_{MTL}(\%)$
Full	Fixed [15]	36.3	61.6	36.1	60.9	89.3	23.8	52.6	-1.65
	Uncertainty [21]	36.2	61.6	35.5	61.2	**89.5**	**24.6**	52.9	-0.76
	GradNorm [5]	23.4	40.9	22.8	25.8	51.3	13.0	28.4	-46.24
	VE-Prompt (Ours)	**39.2**	**64.9**	**39.0**	**63.2**	89.4	24.0	**54.0**	**+1.52**
Disjoint-normal	Fixed [15]	32.6	54.6	32.3	59.7	88.2	23.0	50.9	+1.19
	Uncertainty [21]	32.2	54.1	31.5	59.8	**88.6**	23.8	51.1	+1.79
	GradNorm [5]	25.9	43.2	26.1	39.2	39.6	3.7	27.1	-46.18
	MGDA [41]	25.9	44.6	26.0	50.1	85.4	**25.2**	46.7	-7.26
	VE-Prompt (Ours)	**34.2**	**56.9**	**33.9**	**62.2**	88.3	23.3	**52.0**	**+3.95**
Disjoint-balance	Fixed [15]	31.3	52.8	30.8	60.2	87.0	22.2	50.2	+1.87
	Uncertainty [21]	31.2	53.1	30.9	59.9	87.0	22.2	50.1	+1.66
	GradNorm [5]	28.9	49.0	28.7	46.8	57.4	19.6	38.2	-17.26
	GradNorm* [5]	30.7	51.8	30.4	56.6	86.9	21.7	49.0	-0.73
	MGDA [41]	21.0	38.0	20.3	45.5	82.7	**24.3**	43.4	-12.48
	VE-Prompt (Ours)	**33.9**	**56.6**	**33.7**	**61.2**	**87.4**	22.2	**51.2**	**+4.72**

4.4. Optimization

Since there are four different perception tasks in our network, our multi-task loss contains four parts. For object detection, we adopt the same objective as in DINO [59]. For segmentation-based tasks, i.e., semantic segmentation, drivable area segmentation, and lane detection, we employ the same loss as in Semantic FPN [22]. Therefore, the total loss for the multi-task model is formulated as follows:

$$\mathcal{L}_{\text{total}} = \lambda_{det}\mathcal{L}_{\text{det}} + \lambda_{sem}\mathcal{L}_{\text{sem}} + \lambda_{driv}\mathcal{L}_{\text{driv}} + \lambda_{lane}\mathcal{L}_{\text{lane}}, \quad (6)$$

where $\mathcal{L}_{\text{det}}$, $\mathcal{L}_{\text{sem}}$, $\mathcal{L}_{\text{driv}}$, $\mathcal{L}_{\text{lane}}$ represent objectives for object detection, semantic segmentation, drivable area segmentation, and lane detection, respectively. λ_{det}, λ_{sem}, λ_{driv} and λ_{lane} stand for different loss weights.

5. Experiments

5.1. Dataset Settings

Our experiments are tested on the BDD100K dataset. BDD100K dataset has ∼74k training images and covers both object detection (OD), semantic segmentation (SS), drivable area segmentation (DA), and lane detection (LD). We follow [25] to consider three dataset split settings complying with real-world scenarios, i.e., Disjoint-normal setting, Disjoint-balance setting, and Full setting:

Disjoint-normal Setting The number of labeled images for each task is as follows: object detection (10k), semantic segmentation (7k), drivable area segmentation (20k), and lane detection (20k).

Disjoint-balance Setting There are 21k images in this set and each task has 7k labeled images that are not overlapped with other tasks.

Full Setting Full-setting refers to experimenting on all available annotations on ∼74k images in BDD100K and can be used to analyze the upper bound of different methods.

5.2. Evaluation and Implementation Details

Evaluation Metric In addition to reporting performance on every individual task, we follow [46] to evaluate the whole multi-task performance:

$$\Delta_{MTL} = \frac{1}{T}\sum_{i}^{T}(M_{m,i} - M_{b,i})/M_{b,i}, \tag{7}$$

where $M_{m,i}$ is the performance of multi-task model on task i, and $M_{b,i}$ indicates the result of single-task baseline. Since we choose Sparse R-CNN as the detection head for re-implementing current multi-task methods, we regard it as the baseline for object detection. For object detection, we adopt mAP as the main metric. While for segmentation tasks, we use mIoU to evaluate the model. We also compute the average performance (Avg.) of all tasks to compare experimental results more intuitively.

Implementation Details The default training setting is that epoch and batch size are fixed as 36 and 16, the learning rate is set to 1×10^{-5}, and weight decay is 1×10^{-4}. We adopt the AdamW optimizer, for which the warmup length is 1 epoch and the warmup factor is 0.001. We choose Swin-Tiny [33] as the backbone by default. More details are provided in Appendix.

5.3. Comparison of Multi-task Methods

We study the performances of popular existing multi-task methods under three settings on BDD100K. We adopt Sparse R-CNN to construct the detection head for efficiency.

Partial-label Learning As shown in Table 1, pseudo labeling [15] can improve performances, especially in object detection and semantic segmentation compared with zeroing loss [51]. Pseudo-labeling achieves satisfactory performance in all settings.

Task Scheduling As shown in Table 1, three task sampling methods (i.e., Uniform sampler [26], Weighted sampler [26] and Round-robin [26]) perform better than Zeroing loss [51] by a large margin on segmentation-based tasks, but get worse in object detection. We hypothesize that training one task per step may lead to forgetting to some extent.

Task Balancing We choose pseudo labeling as the baseline since task-balancing methods are more suitable in settings with complete labels. Fixed denotes fixed loss weights for all tasks during training. As shown in Table 2, Uncertainty performs better than Fixed on semantic segmentation and drivable area segmentation under the full and disjoint-normal settings, while performances of other approaches (i.e., GradNorm and MGDA) degrade significantly. Especially, GradNorm uses the last shared layer of weights to compute gradient norm in its paper, thus we adopt the last layer of P_5 in the neck. When we use the full image encoder to compute the gradient norm, GradNorm achieves much better results but still lags behind the baseline. Interestingly, MGDA achieves the best result on lane detection, indicating that it suffers from heavy negative transfer.

For efficiency and effectiveness, we choose pseudo labeling with fixed loss weights as our baseline, which achieves competitive performance compared with other complicated multi-task methods, to verify the effectiveness of VE-Prompt.

5.4. Compare VE-Prompt with Previous Methods

As shown in Table 3, our VE-Prompt surpasses the baseline consistently on almost all metrics in all three settings and achieves significant overall multi-task performance. We conclude that VE-Prompt can learn high-quality task-specific knowledge during training, and further improve performance. VE-Prompt also achieves the best results on three tasks compared with single-task models. We also compare VE-Prompt with LV-Adapter [25] in Appendix.

5.5. Ablation Study

We conduct all ablation studies under the disjoint-balance setting for efficiency.

Module Components We present detailed comparisons on each module to validate our VE-Prompt as in Table 4. The introduced shared transformer encoder alleviates this imbalance to some extent (row 1 vs. row 2). Equipped with task-specific prompts through task prompting, the model gets better results on all tasks (row 2 vs. row 3), confirming that task-specific prompts can motivate the model to learn useful task-specific knowledge for specific tasks.

Task Prompting We conduct an ablation study to compare the proposed two prompting strategies as in Table 5. We can see that both post-head and pre-head prompting improve the performance of object detection. However, post-head prompting gets inferior results on segmentation-based tasks (#1 vs. #2), indicating that the post-head process is not suitable for dense prediction tasks. On the contrary, pre-head prompting helps the model make full use of task-specific knowledge and improves all tasks consistently (#3).

Prompt Initialization The task-specific prompts are initialized with the pre-trained image encoder of CLIP. We compare it with random initialization as in Table 6. Results show that prompts with CLIP initialization improve the multi-task model on all metrics. More comparisons and analyses on prompting are presented in Appendix.

Table 3. Comparison between single-task and multi-task learning baselines under different settings.

Setting	Methods	mAP	AP50	AP75	mIoU (SS)	mIoU (DA)	IoU (LD)	Avg.	$\Delta_{MTL}(\%)$
Full	Sparse R-CNN [43]	36.5	61.5	36.1	-	-	-	-	-
	DINO [59]	38.6	64.2	38.2	-	-	-	-	-
	Semantic FPN [22]	-	-	-	59.8	-	-	-	-
	Semantic FPN [22]	-	-	-	-	89.1	-	-	-
	Semantic FPN [22]	-	-	-	-	-	25.9	-	-
	Sparse R-CNN based	36.3	61.6	36.1	60.9	89.3	23.8	52.6	-1.65
	DINO based	**39.4**	64.5	**39.8**	61.5	84.9	22.0	52.0	-2.25
	VE-Prompt (Ours)	39.2	**64.9**	39.0	**63.2**	**89.4**	**24.0**	**54.0**	**+1.52**
Disjoint-normal	Sparse R-CNN [43]	28.8	50.4	28.0	-	-	-	-	-
	DINO [59]	31.2	53.0	30.5	-	-	-	-	-
	Semantic FPN [22]	-	-	-	59.8	-	-	-	-
	Semantic FPN [22]	-	-	-	-	87.8	-	-	-
	Semantic FPN [22]	-	-	-	-	-	25.2	-	-
	Sparse R-CNN based	32.6	54.6	32.3	59.7	88.2	23.0	50.9	+1.19
	DINO based	33.1	55.9	32.2	59.2	87.2	22.7	50.6	+0.83
	VE-Prompt (Ours)	**34.2**	**56.9**	**33.9**	**62.2**	**88.3**	**23.3**	**52.0**	**+3.95**
Disjoint-balance	Sparse R-CNN [43]	28.1	49.2	26.7	-	-	-	-	-
	DINO [59]	29.4	50.8	28.1	-	-	-	-	-
	Semantic FPN [22]	-	-	-	59.8	-	-	-	-
	Semantic FPN [22]	-	-	-	-	85.5	-	-	-
	Semantic FPN [22]	-	-	-	-	-	23.7	-	-
	Sparse R-CNN based	31.3	52.8	30.8	60.2	87.0	**22.2**	50.2	+1.87
	DINO based	33.5	55.6	33.1	58.1	85.2	21.4	50.0	+1.58
	VE-Prompt (Ours)	**33.9**	**56.6**	**33.7**	**61.2**	**87.4**	**22.2**	**51.2**	**+4.72**

Table 4. Ablation study of modules in our proposed VE-Prompt. TE means transformer encoder.

	mAP	mIoU (SS)	mIoU (DA)	IoU (LD)
DINO based	33.5	58.1	85.2	21.4
w/ shared TE	32.2	60.5	86.5	21.4
+ Prompt	**33.9**	**61.2**	**87.4**	**22.2**

Table 5. Ablation study of task-specific prompts. Post and Pre indicate post-head prompting and pre-head prompting respectively.

#	Prompt	Post	Pre	mAP	mIoU (SS)	mIoU (DA)
1	✗	✗	✗	32.2	60.5	86.5
2	✓	✓	✗	33.2	58.9	86.4
3	✓	✗	✓	**33.9**	**61.2**	**87.4**

Table 6. Ablation study of initialization for prompt vectors.

CLIP Initialization	mAP	mIoU (SS)	mIoU (DA)	IoU (LD)
✗	33.5	61.0	87.2	21.9
✓	**33.9**	**61.2**	**87.4**	**22.2**

6. Conclusion and Discussion

In this paper, we first provide an in-depth analysis of popular multi-task learning methods under the realistic scenarios of self-driving, which covers four common perception tasks, i.e., object detection, semantic segmentation, drivable area segmentation, and lane detection. We find that existing methods cannot solve all tasks satisfactorily due to the negative transfer. To mitigate the negative transfer, we propose visual exemplar driven task-prompting (VE-Prompt), which incorporates visual exemplars of different tasks to provide high-quality task-specific knowledge. Besides, the proposed framework bridges transformer and convolutional layers for efficient and accurate unified perception in autonomous driving. Experimental results show that VE-Prompt can achieve superior performance on large-scale driving dataset BDD100K.

Limitations Although our method has achieved substantial improvement in the overall multi-task metric, we find the model gets worse results on lane detection compared with the single-task model. We conjecture that it is because the lane detection task is quite different from other tasks, thus it is difficult for the multi-task model to solve all four tasks satisfactorily.

Acknowledgements

We gratefully acknowledge the support of MindSpore[1], CANN (Computer Architecture for Neural Networks) and Ascend AI Processor used for this research.

[1] https://www.mindspore.cn/

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901, 2020. 2

[2] David Brüggemann, Menelaos Kanakis, Anton Obukhov, Stamatios Georgoulis, and Luc Van Gool. Exploring relational context for multi-task dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15869–15878, 2021. 1, 2

[3] Rui Chen, Haizhou Ai, Chong Shang, Long Chen, and Zijie Zhuang. Learning lightweight pedestrian detector with hierarchical knowledge distillation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1645–1649. IEEE, 2019. 2

[4] Yaran Chen, Dongbin Zhao, Le Lv, and Qichao Zhang. Multi-task learning for dangerous object detection in autonomous driving. *Information Sciences*, 432:559–571, 2018. 2

[5] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR, 2018. 3, 6

[6] Zhiyang Chen, Yousong Zhu, Zhaowen Li, Fan Yang, Wei Li, Haixin Wang, Chaoyang Zhao, Liwei Wu, Rui Zhao, Jinqiao Wang, et al. Obj2seq: Formatting objects as sequences with class prompt for visual tasks. *arXiv preprint arXiv:2209.13948*, 2022. 2, 3

[7] Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D Manning, and Quoc V Le. Bam! born-again multi-task networks for natural language understanding. *arXiv preprint arXiv:1907.04829*, 2019. 2

[8] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020. 2

[9] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012. 3

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 5

[11] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2650–2658, 2015. 2

[12] Christopher Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 34, 2021. 1, 2

[13] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021. 3

[14] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3205–3214, 2019. 3

[15] 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 (ICCV)*, pages 8856–8865, 2021. 2, 3, 6, 7

[16] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[18] Yun He, Steven Zheng, Yi Tay, Jai Gupta, Yu Du, Vamsi Aribandi, Zhe Zhao, YaGuang Li, Zhao Chen, Donald Metzler, et al. Hyperprompt: Prompt-based task-conditioning of transformers. In *International Conference on Machine Learning*, pages 8678–8690. PMLR, 2022. 2

[19] Keishi Ishihara, Anssi Kanervisto, Jun Miura, and Ville Hautamaki. Multi-task learning with attention for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2902–2911, 2021. 2

[20] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022. 1, 2, 3

[21] Alex Kendall, Yarin Gal, and Roberto Cipolla. 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. 1, 2, 3, 6

[22] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6399–6408, 2019. 2, 4, 6, 8

[23] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6129–6138, 2017. 1, 3

[24] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 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. *arXiv preprint arXiv:2209.08953*, 2022. 6, 7

[26] Valerii Likhosherstov, Anurag Arnab, Krzysztof Choromanski, Mario Lucic, Yi Tay, Adrian Weller, and Mostafa Dehghani. Polyvit: Co-training vision transformers on images, videos and audio. *arXiv preprint arXiv:2111.12993*, 2021. 1, 2, 6, 7

[27] Valerii Likhosherstov, Anurag Arnab, Krzysztof Choromanski, Mario Lucic, Yi Tay, Adrian Weller, and Mostafa Dehghani. Polyvit: Co-training vision transformers on images, videos and audio. *arXiv preprint arXiv:2111.12993*, 2021. 3

[28] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 4

[29] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. *Advances in neural information processing systems*, 32, 2019. 3

[30] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021. 2

[31] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1871–1880, 2019. 3

[32] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *arXiv preprint arXiv:2103.10385*, 2021. 3

[33] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 4, 7

[34] Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. *Advances in Neural Information Processing Systems*, 34, 2021. 1

[35] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3994–4003, 2016. 3

[36] Xing Nie, Bolin Ni, Jianlong Chang, Gaomeng Meng, Chunlei Huo, Zhaoxiang Zhang, Shiming Xiang, Qi Tian, and Chunhong Pan. Pro-tuning: Unified prompt tuning for vision tasks. *arXiv preprint arXiv:2207.14381*, 2022. 3

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021. 3, 5

[38] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18082–18091, 2022. 2, 3

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 28, 2015. 2

[40] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4822–4829, 2019. 3

[41] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018. 2, 3, 6

[42] Trevor Scott Standley, Amir Roshan Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning (ICML)*, pages 9120–9132, 2020. 2

[43] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, and Ping Luo. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14454–14463, 2021. 8

[44] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1013–1020, 2018. 2

[45] Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Mti-net: Multi-scale task interaction networks for multi-task learning. In *European Conference on Computer Vision*, pages 527–543. Springer, 2020. 3

[46] 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*, 2021. 7

[47] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018. 2

[48] Ze Wang, Weiqiang Ren, and Qiang Qiu. Lanenet: Real-time lane detection networks for autonomous driving. *arXiv preprint arXiv:1807.01726*, 2018. 2

[49] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. *arXiv preprint arXiv:2204.04799*, 2022. 2

[50] 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, 2

[51] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision (ECCV)*, pages 418–434, 2018. 2, 6, 7

[52] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018. 3

[53] Yutong Xie, Jianpeng Zhang, Chunhua Shen, and Yong Xia. Cotr: Efficiently bridging cnn and transformer for 3d medical image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 171–180, 2021. 4

[54] 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 Conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018. 3

[55] Yangyang Xu, Xiangtai Li, Haobo Yuan, Yibo Yang, Jing Zhang, Yunhai Tong, Lefei Zhang, and Dacheng Tao. Multi-task learning with multi-query transformer for dense prediction. *arXiv preprint arXiv:2205.14354*, 2022. 1, 2

[56] Zhengyuan Yang, Yixuan Zhang, Jerry Yu, Junjie Cai, and Jiebo Luo. End-to-end multi-modal multi-task vehicle control for self-driving cars with visual perceptions. *24th International Conference on Pattern Recognition (ICPR)*, pages 2289–2294, 2018. 1

[57] Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Cpt: Colorful prompt tuning for pre-trained vision-language models. *arXiv preprint arXiv:2109.11797*, 2021. 3

[58] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 2, 3

[59] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 4, 6, 8

[60] Zhenyu Zhang, Zhen Cui, Chunyan Xu, Zequn Jie, Xiang Li, and Jian Yang. Joint task-recursive learning for semantic segmentation and depth estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 235–251, 2018. 3

[61] Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [mask]: Learning vs. learning to recall. *arXiv preprint arXiv:2104.05240*, 2021. 3

[62] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*, 2021. 1, 2, 3

[63] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 2