RepVF: A Unified Vector Fields Representation for Multi-task 3D Perception

Chunliang Li[1*], Wencheng Han[2*], Junbo Yin[1], Sanyuan Zhao[1†], and Jianbing Shen[2],

[1] School of Computer Science, Beijing Institute of Technology
[2] SKL-IOTSC, Computer and Information Science, University of Macau, China
{jbji, sanyuanzhao}@bit.edu.cn
{wencheng256, yinjunbocn}@gmail.com
jianbingshen@um.edu.mo

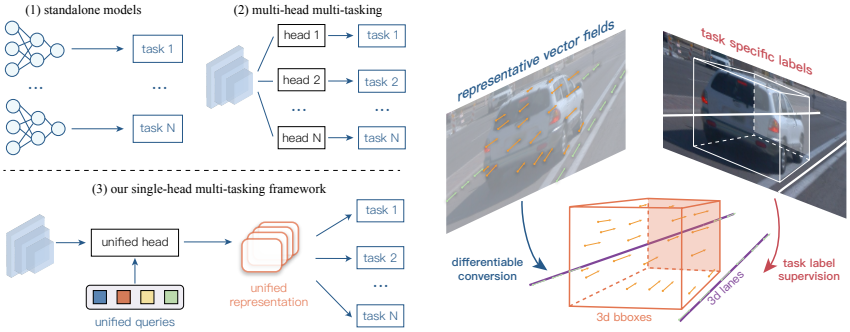
Abstract. Concurrent processing of multiple autonomous driving 3D perception tasks within the same spatiotemporal scene poses a significant challenge, in particular due to the computational inefficiencies and feature competition between tasks when using traditional multi-task learning approaches. This paper addresses these issues by proposing a novel unified representation, RepVF, which harmonizes the representation of various perception tasks such as 3D object detection and 3D lane detection within a single framework. RepVF characterizes the structure of different targets in the scene through a vector field, enabling a single-head, multi-task learning model that significantly reduces computational redundancy and feature competition. Building upon RepVF, we introduce RFTR, a network designed to exploit the inherent connections between different tasks by utilizing a hierarchical structure of queries that implicitly model the relationships both between and within tasks. This approach eliminates the need for task-specific heads and parameters, fundamentally reducing the conflicts inherent in traditional multi-task learning paradigms. We validate our approach by combining labels from the OpenLane dataset with the Waymo Open dataset. Our work presents a significant advancement in the efficiency and effectiveness of multi-task perception in autonomous driving, offering a new perspective on handling multiple 3D perception tasks synchronously and in parallel. The code will be available at: `https://github.com/jbji/RepVF`.

Keywords: 3D Lane Detection · 3D Object Detection · Multi-task Method.

1 Introduction

In autonomous driving, multiple 3D perception tasks often need to be processed synchronously, in real-time, and in parallel. These tasks, which exist within the

*Equal contribution. †Corresponding author. This work was supported in part by the FDCT grants 0102/2023/RIA2, 0154/2022/A3, and 001/2024/SKL, the MYRG-CRG2022-00013-IOTSC-ICI grant and the SRG2022-00023-IOTSC grant.



(a) Multi-tasking paradigm comparison. **(b)** *RepVF*, conversion and supervision.

Fig. 1: *RepVF* is proposed as a unified representation for 3D perception to achieve single-head multi-tasking. It consists of a set of vector fields learned to represent road elements, and is trained utilizing existing labels through differentiable conversions. The proposed single-head multi-tasking paradigm reduces task conflict and competition.

same spatiotemporal scene, are often complementary to each other in both geometric and semantic aspects [19]. For example, 3D object detection [12] and 3D lane detection [11] define foreground objects and background lanes within the same 3D scene. Vehicles usually travel following the guiding semantics of lanes that do not spatially overlap and are partially obscured by other vehicles. Some previous industrial solutions [1, 2] utilize a multi-model paradigm for multi-task perceptions, as depicted in Figure 1a(1). These separate models cannot share common features across different tasks, resulting in computational waste. Furthermore, this paradigm does not take advantage of the inherent connections between different tasks. Some works [19, 28, 38, 39] attempt to overcome this limitation by using a common backbone and separate head networks for each task, as shown in Figure 1a(2). This approach can considerably reduce the computational cost when handling a large number of tasks, but it might lead to a task balancing issue. Since different task-aware heads focus on various aspects of the scene, they may compete for the backbone features, resulting in instability during training [19, 39].

The competition is attributed to the variation in representation between different tasks. For example, the 3D detection task [12, 26, 28, 37, 38, 60, 62] guides the head network to predict 3D bounding boxes, whereas the lane detection task [6, 11] necessitates the corresponding head to regress lane shapes. Specifically, 3D bounding boxes describe a cubic area in 3D space, representing different category foregrounds using position, size, and orientation. In contrast, 3D lanes are represented using anchor [6, 11, 18] or parametric [15] methods to describe one or more 1-D lines. These representations focus on different aspects of the scenes, which ultimately leads to competition.

In this paper, we propose a unified representation, *RepVF* (Representative Vector Fields), for perception tasks in autonomous driving. Using this represen-

tation, we design a single-headed, unified model that can perform tasks simultaneously. As illustrated in Fig. 1b, *RepVF* represents 3D vector fields that each assigns vectors to spatial locations, denoted as $(S, \mathcal{F}(S)) \subseteq \mathbb{R}^{3+d}$ in Sec. 3.2. Sub-vector fields representing different targets in the 3D perception scene progressively adhere to their spatial extent, accurately characterizing the anisotropic structure in their respective spatial vicinity. *RepVF* can be differentiably transformed into task-specific elements, hence requiring no special supervision and can utilize existing labels. With *RepVF*, our network only processes one single type of fundamental perception element. As shown in Fig. 1a, unlike the traditional multi-task approach that employs multiple task heads [19, 28, 30, 38, 72], we follow an unprecedented single-head multi-task paradigm. The single-head multi-task eliminates task-specific parameters and obviates the need to explicitly model task interactions.

RepVF representation serve as a geometrical, cross-scale 3D perception representation alternative. Unlike perception elements capable of handling only single-scale tasks [6, 11, 12, 62], it breaks down the scene into smaller perceptual units while retaining the capability for target-level perception at larger scales. Based on the *RepVF* representation, we further propose RFTR to leverage this relationship, utilizing queries with a hierarchical structure that implicitly models the relationships both between and within different tasks. Following the single-head multi-task architecture, with only shared parameters across tasks, RFTR does not depend on existing multi-task optimization strategies [21, 33, 35, 52, 73]. The gradient discrepancies between tasks across different training iterations have been mitigated, fundamentally reducing the competition and conflict inherent in multi-tasking 3D perception. By employing *RepVF* to replace the fundamental elements for different tasks, and by combining 3D lane labels from the OpenLane dataset [6] with the Waymo Open dataset [53] for simultaneous multi-task training and inference, our approach achieves comparable or even better multi-task performance than single-task counterparts with only one unified model.

In summary, we introduce *RepVF* as a unified representation for multiple perception tasks in autonomous driving, effectively reducing the competition among different task representations during training and enhancing model convergence. We further develop a single-head multi-task framework RFTR based on *RepVF*, fully exploiting the intrinsic connections among different tasks within a scene, improving feature interaction with queries, and enabling gradient-balanced multi-task training. Our approach achieves impressive performance across two key tasks in autonomous driving—3D object detection and 3D lane detection—showcasing the advantages of a unified task representation in the context of autonomous driving applications.

2 Related Work

2.1 Camera-based 3D Perception in Autonomous Driving

3D object detection [7,9,12,62,70] and 3D lane detection [11,14] are two common 3D perception tasks, and previous approaches usually perform the two tasks

independently. 3D object detection has been extensively studied either in model architectures [7, 37, 38, 59] or feature representations [17, 25, 27, 28, 49, 50, 71, 76]. Early methods [20, 22, 45] predict 3D bounding boxes using 2D detectors, and Mono3D [7] is one of the first methods directly make predictions from 3D representations. Motivated by DETR [11], DETR3D [62] uses sparse 3D object centers as queries, and PETR [37] further enhances its feature representations by embedding 3D positions. Some recent approaches [17, 28, 38, 58] also focus on incorporating temporal information into 3D object detection, but they all follow the transformer-based [56] paradigm introduced by DETR3D [62].

Only recently have DETR-based methods been introduced for 3D lane detection [3, 38, 42, 63, 64]. Previous methods [6, 11, 14, 34, 57] prefer to project image features into BEV space using Inverse Perspective Mapping (IPM), but this introduces prohibitive height prediction error. Some work [67] avoids IPM via depth estimation, and Anchor-3DLane [18] reverses this projection process. Following DETR [3, 38, 61] introduce 3d lane anchors as queries, and [42] adopts finer lane prior queries and is end-to-end. With the advances of DETR [61, 62] networks in both tasks [3, 28, 37, 42], it is architecturally ready to unify the two tasks without standalone models.

2.2 Multi-task 3D Perception

The problem of multi-tasking in autonomous driving perception is still in its infancy. Recent advances [19, 28, 37–39, 48, 65, 75] elaborate on universal perception features and multiple tasks are handled with different heads. Moreover, there is no consensus on the required tasks. For example, MMF [29] addresses both depth completion and object detection issues, while [19, 28, 38, 39] performs object detection and map segmentation, with [38] performing additional 3D lane detection separately. However, most of these studies only use multiple tasks to demonstrate the universality of their learned features, and the tasks are trained and evaluated separately [28, 38] rather than simultaneously [19, 39], not to mention dataset discrepancies.

In addition, multitask learning (MTL) typically involves task conflict. Previous MTL studies [55] addresses this either through parameter sharing [13, 44, 47, 51, 54, 66] or optimization [8, 21, 33, 35, 52, 73]. Parameter sharing approaches are conceptually straightforward, and are further classified into hard-sharing (with shared and task-specific parameters) and soft-sharing (with a cross-task talk mechanism). Optimization-based approaches, *e.g.* IMTL [33] and DWA [35], attribute MTL imbalance to the gradient and loss imbalance and explore optimization calibration methods. Taking advantage of these advances, FULLER [19] stands as the pioneering work in autonomous driving perception that analyzes the performance of modern optimization-based MTL calibration methods [8, 33, 35]. With a unified task representation, we follow a different paradigm from previous ones, that eliminate task-specific heads and mitigate the gradient imbalance. Our approach can be seen as a hard-sharing technique without task-specific parameters and does not require calibration.

3 Representative Vector Fields

We begin by revisiting the representations of two common 3D perception tasks: 3D object detection [4, 12, 53] and 3D lane detection [6, 67]. By exploring how task-specific representations have evolved into task-specific designs [3, 6, 7, 11, 37, 38, 42, 50, 62] , we propose the task-agnostic *RepVF* (Representative Vector Fields), which arise from the geometric commonality in the tasks and effectively reduce task competition via unified representations.

3.1 Task-specific Representations and Single-tasking

Representation Formulations. Task-specific representations vary due to the geometric features of the task targets, which differ in dimensions and scales. As a key task for understanding 3D space, 3D object detection deals with objects irregularly shaped and scaled from one to several metres, defined by coarse outer boundaries as 3D bounding boxes [12] , formally a 7-d coordinate $B = (x, y, z, l, w, h, \theta)$ encoding object center (x, y, z), box dimensions (l, w, h) and heading θ. Conversely, due to their linear shape and ability to cover long distances, 3d lanes are not described by boundaries but by direct geometries, typically ordered sequences of N_L 3D point coordinates [6]:

$$L = [(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_{N_l}, z_{N_l}, z_{N_l})] \tag{1}$$

This may look familiar to an unordered point cloud, but they are ordered along the direction of the road. In addition, 3D perception problems typically involve classifications that require category scores $\mathcal{C}_i$, varying from 1, 3 or 23 [4, 12, 53] classes (3D object) to 21 or 22 [6] classes (3D lane).

Evolving into Task-specific Designs. Pioneering 3D object detection from RGB images is proposal based, *e.g.* Mono3D [7] reduces proposal space dimension of each class to (x, y, z, θ, t) with representative templates, and only predicts Δ_{pos}; OFT-Net [50] predicts offsets of all bounding box parameters relative to the top-view grid, *i.e.* $\Delta_{\text{pos}}, \Delta_{\text{dim}}, \Delta_{\text{ang}}$. Taking advantage of the center coordinates, DETR-based methods [37, 62] regress Δ_{pos} along with bounding box parameters. However, constrained by the limited spatial extent of center points, extracted features around them capture only local spatial areas [77], failing to capture the finer pose and structure of objects.

3D lines are suitable for anchor-based representations [6, 11, 38], resulting in equally spaced longitudinal lines and anchor-based predictions Δ_{xz}. Anchor representations reduce the representation dimension, but also restrict the degrees of freedom, limiting the maximum model capacity, which has led some works [6, 18] focus on more sophisticated anchor representations. Recent DETR-based methods [3, 38, 42] use anchor coordinates as queries, exploring the full potential of anchors. Unfortunately, since it's not trivial and stable to directly predict parameter-driven curves [10, 15], mainstream methods stick to predicting anchor-based offsets Δ_{xz}, and nearly no one ever worked directly on the raw lane format.

Intersection of Task-specific Representations. We believe that it's already been architecturally ready to unify the two (and possibly more) perception tasks, with the population and success of DETR practices [19,28,38,61]. However, they still rely on multi-head design due to the discrepancy in task representation, which consumes more computational resources. Despite the significant differences in the dynamics and scales of the described targets, these task-specific representations only encapsulate descriptions of shapes, structures, and directions (object heading and lane ordinality). It is plausible to represent these task targets universally, taking into account perception hierarchies.

3.2 The Task-Agnostic Representation through RepVF

To facilitate efficient design in multitasking frameworks, we introduce *RepVF*, a vector fields representation inspired by 2D point set object representation [43,43]. Distinct from conventional isotropic points, *RepVF* assigns a vector to each location making itself anisotropic. We define *RepVF* as a collection of vector fields, each field denoted by R_i as a perception target, whose semantic vector assignment mapping $\mathcal{F}_i$ is defined on $D = \bigcup_{i=1}^{n_s} S_i \subseteq \mathbb{R}^3$:

$$\{R_i\}, \quad R_i = (S_i, \mathcal{F}_i(S_i)) \subseteq \mathbb{R}^{3+d}, \quad \mathcal{F} : D \to \mathbb{R}^d \tag{2}$$

where d is the dimension of the target vector domain. Therefore, for any point x within D, $\mathcal{F}$ assigns it a vector $\mathcal{F}(x) \in \mathbb{R}^d$ that encapsulates dimensional and positional attributes of the point. This definition allows us to cover the whole space: The sampling domain D is partitioned into n_s disjoint subsets S_i, with each subset containing n representative sampling points: $S_i = \{(x_k, y_k, z_k)\}_{k=1}^{n}$. Accordingly, the collective vector field $\mathcal{F}$ sprawls over $D = \bigcup_{i=1}^{n_s} S_i$, articulated into n_s sub-vector fields $\mathcal{F}_i$ catering to specific tasks.

Intuitively, S_i directly outlines entity-level geometric structures, and $\mathcal{F}_i(S_i)$ delineates contextual characteristics (*i.e.* θ for bounding box B, or adjacent point directions for lane L). Considering the directionality, we use **unit-length** vectors parallel to the x-y plane to capture angular behavior, resulting in behavior akin to scalars and $d = 1$. To represent proper $\mathcal{F}_i$ for 3d lanes, directions of the penultimate points are employed for terminal lane line points.

Converting RepVF into Task-Specific Representation. To leverage existing task labels for training and evaluation, similar to object detection representations based on point sets [43,68], we convert predictions *RepVF* $\hat{R}_i$, category $\hat{\mathcal{C}}_i$ to task-specific representations. We define **geometric interpretation** processes as differentiable, parameter-free transformation functions:

$$\mathcal{T}^{(\cdot)} : \hat{R}_i \to \hat{P}_i, \quad \text{s.t. } \hat{P}_i \in \{\hat{B}, \hat{L}\}_i \tag{3}$$

Specifically, we have $\mathcal{T}^l : \hat{R}_i \to \hat{L}_i$ for 3d lanes and $\mathcal{T}^b : \hat{R}_i \to \hat{B}_i$ for 3d objects. For **semantic parts**, to align $\hat{\mathcal{C}}_i$ with task-specific classes, we split it into $n_{(\cdot)}$ bins and take max score within each bin, where $n_{(\cdot)}$ is the count of task categories. $\hat{R}_i$ and $\hat{R}_j$ remain formally equivalent until contextualized into $\hat{P}_i$ and $\hat{P}_j$, thus maintaining task agnosticism.

Conversion Function Choice. In our implementation, $\mathcal{T}^l$ sorts the unordered points of $\hat{S}_i$ by the road direction($+X$ in Waymo [53]). Unlike 2D boxes, 3D boxes are not strictly aligned with to XYZ axes due to headings. Therefore, $\mathcal{T}^b$ consists *three parts* that respectively compute heading, box center and dimensions:

$$\mathcal{T}^b_h : \hat{\mathcal{F}}_i \to \hat{\theta}_i; \quad \mathcal{T}^b_{xyz} : \hat{S}_i \to (\hat{x}_i, \hat{y}_i, \hat{z}_i); \quad \mathcal{T}^b_{lwh} : \hat{S}_i \to (\hat{l}_i, \hat{w}_i, \hat{h}_i) \tag{4}$$

Here, $\mathcal{T}^b_h$ and $\mathcal{T}^b_{xyz}$ are simple arithmetic averages. With $\hat{\theta}_i$ given by $\mathcal{T}^b_h$, by projecting $\hat{S}_i$ to the main orientation axis $\hat{\theta}_i$, the predictions are aligned with the predicted directions. We then proceed with two candidates of $\mathcal{T}^b_{lwh}$ similar to 2D practices [68]: *(1)* $\mathcal{T}^b_{lwh} = \mathcal{T}_1$, the **min-max function**, performing min/max operations projection dimensions for dimensions;*(2)* $\mathcal{T}^b_{lwh} = \mathcal{T}_2$ employs a **momentum-based function** taking the variance of predicted points as the boundary box dimensions. In our practice, $\mathcal{T}_2$ works slightly better.

Learning RepVF. RepVF is supervised by loss functions universally across tasks without special losses. Here we use L^1 regression loss on S_i and $\mathcal{F}_i$, and focal loss [32] on $\mathcal{C}_i$. Smaller structures are described by points in S_i, and R_i itself is set-level, extending RepVF across different perception hierarchies. To capture this cross-level property, we adopt DETR-based structure [37,61,62] and aligned query design to use raw representations motivated by [3, 36]. With multiple decoder layers [62, 77], the learning process of RepVF can be characterized as *iterative* update Ω_l on queries Q_l and inferring $\hat{R}_l, \hat{\mathcal{C}}_l$ from them with linear layer Φ and decoder layer l:

$$Q_l = \Omega_l(Q_{l-1}), \quad \hat{R}_l, \hat{\mathcal{C}}_l = \Phi(Q_l) \tag{5}$$

We will explain our network in detail in the next section.

4 RFTR: The Single-head Multi-tasker

In this section, we introduce the single-head multi-task framework RFTR (Representative Vector Fields Transformer), which uses *RepVF* representation to unify different tasks and exploit intrinsic connections among them. We also observed one interesting behavior of RFTR in the balance of multi-task gradients. RFTR is built upon DETR networks [5, 37, 62] and is end-to-end on all tasks.

4.1 Network Structure

Figure 2 shows our RFTR architecture built upon *RepVF*. It inputs images I from N cameras with known camera parameters, and outputs unified representative vector fields $\mathcal{F}$ for various perception tasks in the 3D scene. RFTR contains four main components that extract features from images, generate set-level queries for hierarchical representation structure, solve unified representative vector fields and multiple tasks with one single head, and convert results to task representations in a differentiable way to utilize existing task labels.

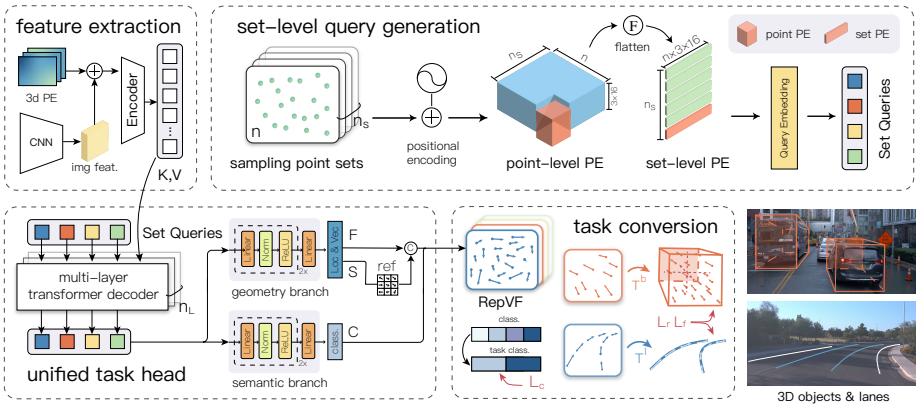


Fig. 2: Overview of our RFTR (Representative Vector Fields Transformer) built upon *RepVF*. Multi-view image features with 3D position embeds are extracted by image backbone and fed into the decoder. We generate set-level query embeds from 3D space sampling sets, each representing a perception target. One single unified task head is then used to predict unified representative vector fields. Finally, predictions are transformed in a differentiable manner into task specific representations to utilize existing labels for supervision.

Feature Extraction. Following common practices in utilizing 2D backbones to extract features [28, 37, 62], we adopt ResNet [16] (ResNet-50) and FPN [31] to produce 2D features F^{2d}. They're encoded with 3D position embedding in PETR [37] into 3D position aware features F^{3d}.

Generating Set-level Perception Queries. As *RepVF* is sampled from n_s point sets S_i with n points each and is two-level, our aim is to generate queries that are consistent with this hierarchical point-set structure. Hence provide the decoder with sufficient spatial information, improving feature interactions. Unlike former approaches [37, 38, 42, 62] that encode one single 3D location, each query in RFTR encodes multiple locations simultaneously.

We generate set-level queries $Q_0 \in \mathbb{R}^{n_s \times d_q}$ from initial point sets corresponding to S_i, where d_q is the dimension of each query. The generation process $\mathcal{G}$ consists of encoding initialized positions from the point level, and embedding them at the set level:

$$\mathcal{G} = \text{embed}(\text{Flatten}(\text{PE})) \tag{6}$$

where PE is the positional encoding [56]. Point-level PE is flattened into set-level and embedded by a small MLP with two layers. To avoid confusing the model with expansive locations falling on multiple targets in the early stages of training, we use n identical 3D locations sampled with 0-1 uniform distribution for all n_s initial point sets.

To our best knowledge, the closest generation approach to ours is [3], which generates query from an ordered sequence of anchor points motivated by [36].

However, ours is from an unordered sampling point set and is not task-specific. Moreover, we use a tiny 16-dimensional PE for each coordination, divert from former methods [37, 62] with a much larger 256-dimension. In our multi-task setting, queries are equivalently initialized, thus task-agnostic. We divide the first 80% as 3D objects and the last 20% as 3D lanes.

Single-head Multi-tasking. Similar to all previous approaches [5, 37, 62], queries are learnable and updated in decoder layers *iteratively*. The unified head Φ of RFTR is composed by a multi-layer transformer decoder [56], and two branches Φ_g and Φ_s dedicated for geometric regression and semantic prediction respectively to predict *RepVF*. We use the same $n_L = 6$ decoder layers with previous methods [5,37,62] to exploit intrinsic tasks connections implicitly. The interaction process within these layers can be formally expressed as:

$$Q_l = \Omega_l(F^{3d}, Q_{l-1}), \quad l = 1, \ldots, N_L \tag{7}$$

With unified head $\Phi = \{\Phi_g, \Phi_s\}$, predictions $\hat{R}_{li}$ and $\hat{\mathcal{C}}_{li}$ is decoded from i-th query at l-th layer:

$$\hat{R}_{li} = \Phi_g(Q_{li}), \quad \hat{\mathcal{C}}_{li} = \Phi_s(Q_{li}) \tag{8}$$

For Φ_g, additional offsets are added to produce S_{li}. Unified geometric predictions $\hat{R}_l = (\hat{S}_l, \hat{F}_l) \subset \mathbb{R}^{n_s \times (3+d_p)}$ are converted to task-specific representations:

$$\hat{L}_{li} = \mathcal{T}^l(\hat{R}_{li}), \quad \hat{B}_{li} = (\mathcal{T}^b_{xyz}, \mathcal{T}^b_{\theta}, \mathcal{T}^b_{lwh})(\hat{R}_{li}) \tag{9}$$

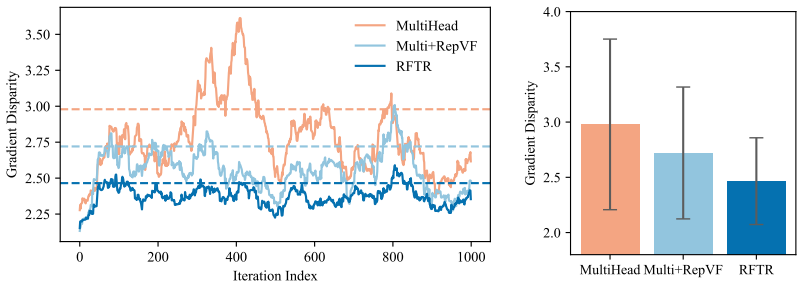
Here, $\mathcal{T}^{(\cdot)}$ are differentiable functions predefined in section 3.2. Each head branch $\Phi_{(\cdot)}$is a simple small MLP network with three fully connected layers. Following DETR3D [62], we weight the loss for the predictions from each decoder layer Ω_l during training, and only outputs from the last layer is used during inference.

Loss Function. RFTR uses existing task labels for supervision, and its loss is calculated between converted predictions and task ground truths. It consists of the following components for coordinate regression, vector fields regression and classification respectively:

$$\mathcal{L}^{(\cdot)} = w_r^{(\cdot)}\mathcal{L}_r^{(\cdot)} + w_f^{(\cdot)}\mathcal{L}_f^{(\cdot)} + w_c^{(\cdot)}\mathcal{L}_c^{(\cdot)}, \quad \mathcal{L} = \sum_{\text{tasks}} \mathcal{L}^{(\cdot)} \tag{10}$$

where $w^{(\cdot)}$ represent loss component weights for task $(\cdot)$, *i.e.* 3D object detection and 3D lane detection. We use the Hungarian algorithm [23] for label assignment after cost calculation between ground truth and converted predictions $\hat{B}_{li}, \hat{P}_{li}$.

In practice, we choose $w_c^b = w_c^l = 2.0$, $w_r^l = 0.003$, $w_r^b = 0.1$, $w_f^b = 0.2$, and $w_f^l = 0.032$. Coordinate regression and classification weights are selected empirically to equalize the scales and contributions of each loss term [74], fields weights are empirically selected. We use L^1 loss for R_i regression, and focal loss [32] with $\gamma = 2.0$ and $\alpha = 0.25$ for classification. Since we predict a degenerated scalar field, we supervise the angular prediction by its sine and cosine values. All tasks share the same losses and similar weights, no additional, task-specific loss measures are adopted.



(a) Gradient disparities by iteration index. **(b)** Average gradient disparities.

Fig. 3: Our single-head RFTR model shows a reduced gradient imbalance and better disparity stability (mean: 2.47, variance: 0.90) compared to the multi-head baseline (mean: 2.98 variance: 1.86) or the multi-head baseline with RepVF (mean: 2.72, variance: 1.57). Ideally balanced gradient disparity through training iterations should approach 2. For ease of trend observation, we have clipped the top 2% values to the mean, with curves smoothed through a window size of 50 1-d convolution.

4.2 Gradient Balance in Single-head Multi-tasking

The multi-head multi-task approach often leads to task conflict, attributed in prior studies [21, 33, 46, 52] to gradient imbalance. These studies [33, 46] and more recent work [19] have explored appropriate gradient calibration strategies to address this issue. As we will show, our single-head RFTR naturally achieves better balanced multi-task gradients thus calibration may no longer be required.

To elucidate the advantage of our single-head multi-task paradigm over multi-head multi-task settings, we use an improved measure of **gradient disparity** adapted from FULLER [19], by taking into account the symmetry disparity:

$$\text{diff}(\nabla\mathcal{L}^l, \nabla\mathcal{L}^b) = \frac{\|\nabla\mathcal{L}^l\|}{\|\nabla\mathcal{L}^b\|} + \frac{\|\nabla\mathcal{L}^b\|}{\|\nabla\mathcal{L}^l\|} \tag{11}$$

where $\|\cdot\|$ is the Frobenius norm and $\nabla\mathcal{L}^{(\cdot)}$ denotes gradients. Given that $\|\nabla\mathcal{L}^{(\cdot)}\|$ is always greater than 0, the value of *diff* function is always greater than or equal to 2. Therefore, the more stable and closer to 2 the gradient disparity is during model training, the better the multi-task stability indicated.

Figure 3a illustrates the gradient disparity curves through training iterations. The multi-head multi-task method [37, 38] is under the same task settings with ours. In the sampling 1k iterations, the stability and absolute scale of the gradient disparity curve for the multi-head configuration are inferior to our single-head multi-task structure. The mean and variance of the multi-head disparities are about 108.5% and 107.8% higher, respectively, than our single-headed RFTR approach, as shown in Fig. 3b. We have also observed that unified representation and single head design contributed to the improvement about equally. More importantly, this means that our single-head RFTR *mitigates*, though not eliminating, the gradient imbalance between tasks, which is fundamentally different from task gradient balancing strategies that do not change the gradient itself.

5 Experiments

We evaluate our method on the Waymo Open Dataset [53] extended by 3D lanes annotated in OpenLane [6], so that we train and evaluate tasks simultaneously.

5.1 Datasets

Waymo Open Dataset and OpenLane. Waymo Open Dataset (Perception) [53] is a large-scale multimodal dataset composed of data from 5 cameras(front and sides), 1 mid-range lidar, and 4 short-range lidars. It contains 1000 sequences in total(about 198k samples), a training set of 798 sequences and a validation set of 202 sequences. OpenLane [6] is a comprehensive real-world 3D lane detection dataset built upon the Waymo Open dataset [53]. It includes a vast array of 200K frames and over 880K carefully annotated lanes, making it one of the largest of its kind to date. The dataset encompasses complex lane structures and features a variety of scene tags, such as weather and locations. Designed to emulate real-world scenarios, OpenLane provides a challenging benchmark for advanced lane detection algorithms. We integrate its 3D lane labels with the original Waymo Open sensor data for multi-task annotation.

Alignment of Datasets. To achieve both tasks simultaneously, we align both datasets in terms of coordinate systems and data splits. *(1)* Coordinate systems. For the vehicle frame, OpenLane [6] uses a vehicle coordinate system that corresponds to the "y-front, x-right, z-up" positioning of 3D-LaneNet [11]. We then align this system to coincide with the commonly utilized "x-front, y-right, z-up" axis for lidar-based 3D object detection. With respect to the sensor frame, we apply transformations to the camera intrinsics so that the sensor frames for all cameras conform to a uniform "z-depth, x-right, y-down" frame. The predictions are then transformed back into their respective task-based coordinate systems for evaluation. *(2)* Data splitting. The OpenLane [6] dataset provides a 1000 segment full version and a 300 segment smaller subset (30%). The partitioning of the larger version complements the original train/validation split of the Waymo Open Dataset. The subset version partitioning is aligned with OpenLane. We perform ablation studies on the smaller subset.

5.2 Implementation Details

Data Processing. We use the v1.4.2 of Waymo Open Dataset [53], and v1.2 for OpenLane [6]. We set the perception range to $[-84.88m, 84.88m]$ for the X and Y axis, and $[-10m, 10m]$ for the Z axis. 3D objects that do not fall within the perception range, are invisible to the cameras, or do not have LiDAR points, are filtered. For 3D lanes, visibility filtering is first applied after axis transformation, then 3D lanes outside the perception range are pruned and then resampled to a fixed size. Approximately 2.07% of the frames without annotations after filtering are excluded from use.

Table 1: Comparison of recent models on 3D lane detection using the OpenLane [6] validation set. The down arrow indicates that lower metric values correspond to better performance, and vice versa. Red indicates the best result, and blue the second-best.

Methods	F-Score ↑	Category Accuracy ↑	X error (m) ↓		Z error (m) ↓	
			near	*far*	*near*	*far*
3D-LaneNet [11] [ICCV19]	44.1	-	0.479	0.572	0.367	0.443
Gen-LaneNet [14] [ECCV20]	32.3	-	0.591	0.684	0.411	0.521
PersFormer [6] [ECCV22]	50.5	92.3	0.485	0.553	0.364	0.431
Cond-IPM [6]	36.6	-	0.563	1.080	0.421	0.892
CurveFormer [3] [ICRA23]	50.5	-	0.340	0.772	0.207	0.651
PETRv2-E [38] [ICCV23]	51.9	-	0.493	0.643	0.322	0.463
PETRv2-V-10 [38]	57.8	-	0.427	0.582	0.293	0.421
PETRv2-V-400 [38]	61.2	-	0.400	0.573	0.265	0.413
BEV-LaneDet [57] [CVPR23]	58.4	-	0.309	0.659	0.244	0.631
Anchor3DLane [18] [CVPR23]	53.1	90.0	0.300	0.311	0.103	0.139
SPG [69] [ICCV23]	52.3	-	0.468	0.514	0.371	0.418
RFTR (Ours)	61.8	91.6	0.341	0.450	0.073	0.107

Table 2: Comparison of recent works of 3D object detection on the Waymo Open Dataset [53] val set. mAPs are LEVEL_1 in Waymo metrics.

Methods	Overall		Vehicle		Cyclist		Pedestrian	
	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
DETR3D [62]	10.1	9.6	9.2	9.1	15.8	15.5	5.3	4.1
DETR3D [62]	8.6	8.0	18.1	18.0	2.4	1.6	5.3	4.4
PETR [37]	20.9	19.7	31.1	30.8	19.5	17.6	12.1	10.6
RFTR (Ours)	19.5	18.7	22.6	22.4	27.8	26.3	8.0	7.2

Model and Training. We used ResNet 50 [16] as the 2D image backbone for fair comparisons. Following PETR [37], the same upsampled and fused P4 feature with 1/16 input resolution is used as the 2D feature. A CPFPN [31] is used as the image neck. All 3D coordinates are normalized to $[0, 1]$. The same weights are used for both the loss computation and the bipartite matching. We adopt progressive weighting for the loss of each decoder layer to improve the convergence speed. For 3D object detection, we compute matching costs based on the center displacement and class score weightings. Dimensions are refined post-matching using the ground truth headings to increase training robustness. We use a total of $n_s = 750$ queries with $n = 15$ sample locations each for the two tasks, 600 for 3D objects and 150 for 3D lanes. We trained our model using the AdamW [41] optimizer with a 0.01 weight decay. The learning rate started at 2.0×10^{-4} and was decayed with a cosine annealing policy [40]. We adopted a multi-scale training strategy, with the shorter side chosen randomly from $[640, 900]$ and the longer side not exceeding 1600. We applied rotation augmentations between $[-\pi/18, \pi/18]$. We resized images of sizes 1920×1080 and 1920×1280 from different views to 1600×900 before augmenting. The final resolution for network inputs was 1408×512. All experiments on the full dataset are trained for 24 epochs on 4 GeForce RTX 4090 GPUs with a batch size of 8. No test time augmentation methods are used during inference.

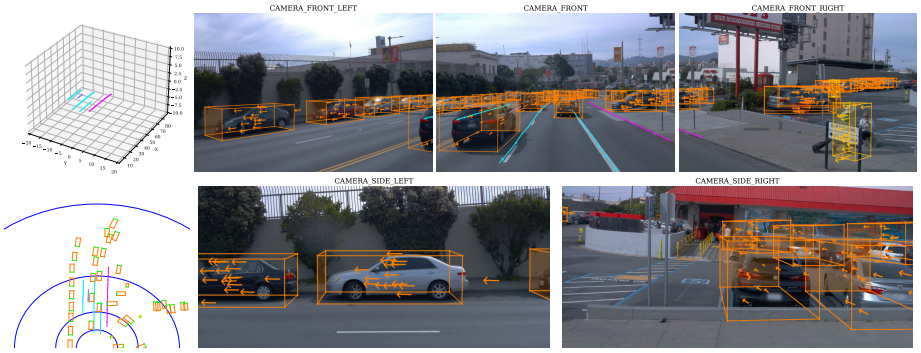


Fig. 4: Visualization of the RFTR model's performance and the learned *RepVF*.

5.3 Comparison with SOTA

3D Lane Detection. As shown in Tab. 1, our RFTR model showcases significant achievements in 3D lane detection on the OpenLane validation set, demonstrating its competitive edge with a high F-score of 61.8 and category accuracy of 91.6%. It matches or exceeds state-of-the-art models such as PETRv2-V-400 [38] (more than 26× denser representation than ours) when assessed on the OpenLane [6] validation set. Notably, while our model has a slightly higher X error in the "far" class than [18], it manages to lead with a higher F-score and near accurate Z errors. Moreover, our method yields notable performance improvements over the Persformer [6] baseline, reflecting the advantageous adaptation of our unified *RepVF* representation and the RFTR architecture.

3D Object Detection. The performance of our RFTR model in 3D object detection, as demonstrated in Tab. 2, shows comparable promising results on the Waymo Open Dataset [53] validation set. While our approach shows a commanding lead in detecting vehicles and cyclists with a solid L1 mAP and mAPH, it shows challenges in detecting smaller entities such as pedestrians. It should be noted, however, that these preliminary results indicate the great potential of using our unified perceptual model for complex tasks, which will require further fine-tuning for object classes with less prominent features.

5.4 Ablation Studies

Head design. We begin by finding out how head designs and representations contribute to the performance, in Tab. 3. The *RepVF* representation with the single-head design, *i.e.* RFTR, performs the best in general with 37.4M parameters. Unified representation brings performance improvements in multi-task learning and reduces task competition. In contrast, task-specific multi-head design produces inferior and unstably biased performance due to task competition, even with more parameters.

Table 3: Multi-task head design and representation ablation. Tasks are trained jointly on 30% data of Waymo Open Dataset (L1 mAP) and OpenLane (F-Score). TS denotes task-specific representation. †: lightweight models.

RepVF	TS	Head	Params↓	F-Score↑	Vehicle↑
	✓	Multi †	37.1M	49.6	20.1
	✓	Multi	46.7M	58.7	17.8
✓		Multi†	39.7M	63.4	18.2
✓		Multi	49.2M	66.4	22.1
✓		Single	37.4M	66.5	25.3

Table 4: Results with different representations on 3D lane. 'Acc' for category accuracy and 'Error' for X error near. MT means simultaneous multi-task 3D lane and 3D object detection.

Representation	F-Score↑	Acc↑	Error↓
Anchor	53.4	86.0	0.476
PS	58.1	88.0	0.545
RepVF	66.0	91.4	0.439
RepVF + MT	66.5	91.7	0.420

Representation. Moreover, we have evaluated the effectiveness of different representations for 3D lane detection in Tab. 4, including traditional anchor representations [6, 37], simple randomly sampled point set representation (PS), and our proposed representative vector fields (RepVF). This study reveals that *RepVF* significantly outperforms traditional anchor and PS methods, and incorporating multi-task learning marginally improves these metrics.

5.5 Qualitative Results

Figure 4 demonstrates the capability of RFTR to simultaneously achieve 3D lane detection and 3D object detection on the extended [24] Waymo open dataset [53]. The figure clearly illustrates that RFTR can accurately predict *RepVF* and perform both tasks at once. In terms of 3D lane detection, RFTR is capable of predicting the shape of lane lines in areas obscured by vehicles, demonstrating its robustness against occlusion. Regarding 3D object detection, the unified RepVF representation accurately captures the 3D bounding boxes. This dual achievement highlights RFTR's effectiveness in handling complex driving scenarios, enhancing both navigation and safety features in autonomous driving systems.

6 Conclusion

In this paper, we propose *RepVF*, a novel unified vector fields representation that characterizes the geometric and semantic structure of different 3D perception targets in the scene. Built upon *RepVF*, our proposed RFTR exploits the intrinsic connections between tasks, mitigates multi-task gradient imbalance and feature competition, and improves model convergence and multi-task expression. Our work not only contributes a novel single-head perspective to the multi-task learning paradigm in autonomous driving but also sets a promising direction for future research in efficient and effective 3D perception task handling.

References

1. Mobileye at CES 2024. https://www.mobileye.com/ces-2024/
2. NVIDIA DRIVE Solutions. https://developer.nvidia.com/drive
3. Bai, Y., Chen, Z., Fu, Z., Peng, L., Liang, P., Cheng, E.: CurveFormer: 3D Lane Detection by Curve Propagation with Curve Queries and Attention. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 7062–7068 (May 2023). https://doi.org/10.1109/ICRA48891.2023.10161160
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A Multimodal Dataset for Autonomous Driving. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11618–11628. IEEE, Seattle, WA, USA (Jun 2020). https://doi.org/10.1109/CVPR42600.2020.01164
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-End Object Detection with Transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 213–229. Lecture Notes in Computer Science, Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_13
6. Chen, L., Sima, C., Li, Y., Zheng, Z., Xu, J., Geng, X., Li, H., He, C., Shi, J., Qiao, Y., Yan, J.: PersFormer: 3D Lane Detection via Perspective Transformer and the OpenLane Benchmark. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022, vol. 13698, pp. 550–567. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-19839-7_32
7. Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R.: Monocular 3D Object Detection for Autonomous Driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2147–2156 (2016)
8. Chen, Z., Badrinarayanan, V., Lee, C.Y., Rabinovich, A.: GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks. In: Proceedings of the 35th International Conference on Machine Learning. pp. 794–803. PMLR (Jul 2018)
9. Cheng, W., Yin, J., Li, W., Yang, R., Shen, J.: Language-Guided 3D Object Detection in Point Cloud for Autonomous Driving (May 2023)
10. Feng, Z., Guo, S., Tan, X., Xu, K., Wang, M., Ma, L.: Rethinking Efficient Lane Detection via Curve Modeling (May 2023). https://doi.org/10.48550/arXiv.2203.02431
11. Garnett, N., Cohen, R., Pe'er, T., Lahav, R., Levi, D.: 3D-LaneNet: End-to-End 3D Multiple Lane Detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2921–2930. IEEE, Seoul, Korea (South) (Oct 2019). https://doi.org/10.1109/ICCV.2019.00301
12. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3354–3361 (Jun 2012). https://doi.org/10.1109/CVPR.2012.6248074
13. Guo, P., Lee, C.Y., Ulbricht, D.: Learning to Branch for Multi-Task Learning. In: Proceedings of the 37th International Conference on Machine Learning. pp. 3854–3863. PMLR (Nov 2020)
14. Guo, Y., Chen, G., Zhao, P., Zhang, W., Miao, J., Wang, J., Choe, T.E.: GenLaneNet: A Generalized and Scalable Approach for 3D Lane Detection. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp.

666–681. Lecture Notes in Computer Science, Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58589-1_40
15. Han, W., Shen, J.: Decoupling the Curve Modeling and Pavement Regression for Lane Detection (Sep 2023)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
17. Huang, J., Huang, G.: BEVDet4D: Exploit Temporal Cues in Multi-camera 3D Object Detection (Jun 2022)
18. Huang, S., Shen, Z., Huang, Z., Ding, Z.h., Dai, J., Han, J., Wang, N., Liu, S.: Anchor3DLane: Learning To Regress 3D Anchors for Monocular 3D Lane Detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17451–17460 (2023)
19. Huang, Z., Lin, S., Liu, G., Luo, M., Ye, C., Xu, H., Chang, X., Liang, X.: FULLER: Unified Multi-modality Multi-task 3D Perception via Multi-level Gradient Calibration. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3479–3488. IEEE, Paris, France (Oct 2023). https://doi.org/10.1109/ICCV51070.2023.00324
20. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1521–1529 (2017)
21. Kendall, A., Gal, Y., Cipolla, R.: Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7482–7491 (2018)
22. Ku, J., Pon, A.D., Waslander, S.L.: Monocular 3D Object Detection Leveraging Accurate Proposals and Shape Reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11867–11876 (2019)
23. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1-2), 83–97 (1955). https://doi.org/10.1002/nav.3800020109
24. Li, L.H., Yatskar, M., Yin, D., Hsieh, C.J., Chang, K.W.: VisualBERT: A Simple and Performant Baseline for Vision and Language (Aug 2019). https://doi.org/10.48550/arXiv.1908.03557
25. Li, X., Yin, J., Li, W., Xu, C., Yang, R., Shen, J.: Di-v2x: Learning domain-invariant representation for vehicle-infrastructure collaborative 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 3208–3215 (2024)
26. Li, X., Yin, J., Shi, B., Li, Y., Yang, R., Shen, J.: Lwsis: Lidar-guided weakly supervised instance segmentation for autonomous driving. Proceedings of the AAAI Conference on Artificial Intelligence **37**(2), 1433–1441 (Jun 2023). https://doi.org/10.1609/aaai.v37i2.25228, https://ojs.aaai.org/index.php/AAAI/article/view/25228
27. Li, Y., Chen, Y., Qi, X., Li, Z., Sun, J., Jia, J.: Unifying Voxel-based Representation with Transformer for 3D Object Detection. Advances in Neural Information Processing Systems **35**, 18442–18455 (Dec 2022)
28. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: BEVFormer: Learning Bird's-Eye-View Representation from Multi-camera Images via Spatiotemporal Transformers. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022. pp. 1–18. Lecture Notes in Computer Science, Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-20077-9_1

29. Liang, M., Yang, B., Chen, Y., Hu, R., Urtasun, R.: Multi-Task Multi-Sensor Fusion for 3D Object Detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7345–7353 (2019)
30. Liang, T., Xie, H., Yu, K., Xia, Z., Lin, Z., Wang, Y., Tang, T., Wang, B., Tang, Z.: BEVFusion: A Simple and Robust LiDAR-Camera Fusion Framework. Advances in Neural Information Processing Systems **35**, 10421–10434 (Dec 2022)
31. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature Pyramid Networks for Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2117–2125 (2017)
32. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal Loss for Dense Object Detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2980–2988 (2017)
33. Liu, L., Li, Y., Kuang, Z., Xue, J.H., Chen, Y., Yang, W., Liao, Q., Zhang, W.: Towards Impartial Multi-task Learning. In: International Conference on Learning Representations (Oct 2020)
34. Liu, R., Chen, D., Liu, T., Xiong, Z., Yuan, Z.: Learning to Predict 3D Lane Shape and Camera Pose from a Single Image via Geometry Constraints. Proceedings of the AAAI Conference on Artificial Intelligence **36**(2), 1765–1772 (Jun 2022). https://doi.org/10.1609/aaai.v36i2.20069
35. Liu, S., Johns, E., Davison, A.J.: End-to-End Multi-Task Learning with Attention (Apr 2019). https://doi.org/10.48550/arXiv.1803.10704
36. Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L.: DAB-DETR: DYNAMIC ANCHOR BOXES ARE BETTER QUERIES FOR DETR (2022)
37. Liu, Y., Wang, T., Zhang, X., Sun, J.: PETR: Position Embedding Transformation for Multi-view 3D Object Detection. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022, vol. 13687, pp. 531–548. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-19812-0_31
38. Liu, Y., Yan, J., Jia, F., Li, S., Gao, A., Wang, T., Zhang, X.: PETRv2: A Unified Framework for 3D Perception from Multi-Camera Images
39. Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D.L., Han, S.: BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 2774–2781 (May 2023). https://doi.org/10.1109/ICRA48891.2023.10160968
40. Loshchilov, I., Hutter, F.: SGDR: Stochastic Gradient Descent with Warm Restarts. In: International Conference on Learning Representations (Nov 2016)
41. Loshchilov, I., Hutter, F.: DECOUPLED WEIGHT DECAY REGULARIZATION (2019)
42. Luo, Y., Zheng, C., Yan, X., Kun, T., Zheng, C., Cui, S., Li, Z.: LATR: 3D Lane Detection from Monocular Images with Transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7941–7952 (2023)
43. Ma, Z., Wang, L., Zhang, H., Lu, W., Yin, J.: RPT: Learning Point Set Representation for Siamese Visual Tracking. In: Bartoli, A., Fusiello, A. (eds.) Computer Vision – ECCV 2020 Workshops. pp. 653–665. Lecture Notes in Computer Science, Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-68238-5_43
44. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-Stitch Networks for Multi-Task Learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3994–4003 (2016)

45. Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J.: 3D Bounding Box Estimation Using Deep Learning and Geometry. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7074–7082 (2017)
46. Navon, A., Shamsian, A., Achituve, I., Maron, H., Kawaguchi, K., Chechik, G., Fetaya, E.: Multi-Task Learning as a Bargaining Game (Feb 2022)
47. Neven, D., De Brabandere, B., Georgoulis, S., Proesmans, M., Van Gool, L.: Fast Scene Understanding for Autonomous Driving (Aug 2017). https://doi.org/10.48550/arXiv.1708.02550
48. Philion, J., Fidler, S.: Lift, Splat, Shoot: Encoding Images from Arbitrary Camera Rigs by Implicitly Unprojecting to 3D. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 194–210. Lecture Notes in Computer Science, Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58568-6_12
49. Reading, C., Harakeh, A., Chae, J., Waslander, S.L.: Categorical Depth Distribution Network for Monocular 3D Object Detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8555–8564 (2021)
50. Roddick, T., Kendall, A., Cipolla, R.: Orthographic Feature Transform for Monocular 3D Object Detection (Nov 2018). https://doi.org/10.48550/arXiv.1811.08188
51. Ruder, S., Bingel, J., Augenstein, I., Søgaard, A.: Latent Multi-Task Architecture Learning. Proceedings of the AAAI Conference on Artificial Intelligence **33**(01), 4822–4829 (Jul 2019). https://doi.org/10.1609/aaai.v33i01.33014822
52. Sener, O., Koltun, V.: Multi-Task Learning as Multi-Objective Optimization. In: Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018)
53. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: Scalability in Perception for Autonomous Driving: Waymo Open Dataset
54. Teichmann, M., Weber, M., Zöllner, M., Cipolla, R., Urtasun, R.: MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving. In: 2018 IEEE Intelligent Vehicles Symposium (IV). pp. 1013–1020 (Jun 2018). https://doi.org/10.1109/IVS.2018.8500504
55. Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., Van Gool, L.: Multi-Task Learning for Dense Prediction Tasks: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(7), 3614–3633 (Jul 2022). https://doi.org/10.1109/TPAMI.2021.3054719
56. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is All you Need. In: Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
57. Wang, R., Qin, J., Li, K., Li, Y., Cao, D., Xu, J.: BEV-LaneDet: An Efficient 3D Lane Detection Based on Virtual Camera via Key-Points. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1002–1011 (2023)
58. Wang, S., Liu, Y., Wang, T., Li, Y., Zhang, X.: Exploring Object-Centric Temporal Modeling for Efficient Multi-View 3D Object Detection (Jun 2023)
59. Wang, T., Zhu, X., Pang, J., Lin, D.: FCOS3D: Fully Convolutional One-Stage Monocular 3D Object Detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 913–922 (2021)

60. Wang, Y., Yin, J., Li, W., Frossard, P., Yang, R., Shen, J.: Ssda3d: Semi-supervised domain adaptation for 3d object detection from point cloud. Proceedings of the AAAI Conference on Artificial Intelligence **37**(3), 2707–2715 (Jun 2023). https://doi.org/10.1609/aaai.v37i3.25370, https://ojs.aaai.org/index.php/AAAI/article/view/25370
61. Wang, Y., Zhang, X., Yang, T., Sun, J.: Anchor DETR: Query Design for Transformer-Based Detector. Proceedings of the AAAI Conference on Artificial Intelligence **36**(3), 2567–2575 (Jun 2022). https://doi.org/10.1609/aaai.v36i3.20158
62. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: DETR3D: 3D Object Detection from Multi-view Images via 3D-to-2D Queries. In: Proceedings of the 5th Conference on Robot Learning. pp. 180–191. PMLR (Jan 2022)
63. Wu, D., Chang, J., Jia, F., Liu, Y., Wang, T., Shen, J.: Topomlp: An simple yet strong pipeline for driving topology reasoning. arXiv preprint arXiv:2310.06753 (2023)
64. Wu, D., Jia, F., Chang, J., Li, Z., Sun, J., Han, C., Li, S., Liu, Y., Ge, Z., Wang, T.: The 1st-place solution for cvpr 2023 openlane topology in autonomous driving challenge. arXiv preprint arXiv:2306.09590 (2023)
65. Xie, E., Yu, Z., Zhou, D., Philion, J., Anandkumar, A., Fidler, S., Luo, P., Alvarez, J.M.: M^{2}BEV: Multi-Camera Joint 3D Detection and Segmentation with Unified Birds-Eye View Representation (Apr 2022). https://doi.org/10.48550/arXiv.2204.05088
66. Xu, D., Ouyang, W., Wang, X., Sebe, N.: 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. pp. 675–684 (2018)
67. Yan, F., Nie, M., Cai, X., Han, J., Xu, H., Yang, Z., Ye, C., Fu, Y., Mi, M.B., Zhang, L.: ONCE-3DLanes: Building Monocular 3D Lane Detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17143–17152 (2022)
68. Yang, Z., Liu, S., Hu, H., Wang, L., Lin, S.: RepPoints: Point Set Representation for Object Detection. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9656–9665. IEEE, Seoul, Korea (South) (Oct 2019). https://doi.org/10.1109/ICCV.2019.00975
69. Yao, C., Yu, L., Wu, Y., Jia, Y.: Sparse Point Guided 3D Lane Detection
70. Yin, J., Fang, J., Zhou, D., Zhang, L., Xu, C.Z., Shen, J., Wang, W.: Semi-supervised 3D Object Detection with Proficient Teachers. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022, vol. 13698, pp. 727–743. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-19839-7_42
71. Yin, J., Shen, J., Chen, R., Li, W., Yang, R., Frossard, P., Wang, W.: Is-fusion: Instance-scene collaborative fusion for multimodal 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14905–14915 (2024)
72. Yin, J., Wang, W., Meng, Q., Yang, R., Shen, J.: A unified object motion and affinity model for online multi-object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6768–6777 (2020)
73. Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., Finn, C.: Gradient Surgery for Multi-Task Learning (Dec 2020). https://doi.org/10.48550/arXiv.2001.06782

74. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking. International Journal of Computer Vision **129**(11), 3069–3087 (Nov 2021). https://doi.org/10.1007/s11263-021-01513-4
75. Zhou, D., Fang, J., Song, X., Liu, L., Yin, J., Dai, Y., Li, H., Yang, R.: Joint 3d instance segmentation and object detection for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1839–1849 (2020)
76. Zhou, D., Song, X., Dai, Y., Yin, J., Lu, F., Liao, M., Fang, J., Zhang, L.: Iafa: Instance-aware feature aggregation for 3d object detection from a single image. In: Proceedings of the Asian Conference on Computer Vision (2020)
77. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable Transformers for End-to-End Object Detection (Mar 2021). https://doi.org/10.48550/arXiv.2010.04159