Towards 3D Object Detection with 2D Supervision

Jinrong Yang\textsuperscript{1} Tiancai Wang\textsuperscript{2} Zheng Ge\textsuperscript{2} Weixin Mao\textsuperscript{2} Xiaoping Li\textsuperscript{1} Xiangyu Zhang\textsuperscript{2}
\textsuperscript{1}Huazhong University of Science and Technology \textsuperscript{2}MEGVII Technology

Abstract

The great progress of 3D object detectors relies on large-scale data and 3D annotations. The annotation cost for 3D bounding boxes is extremely expensive while the 2D ones are easier and cheaper to collect. In this paper, we introduce a hybrid training framework, enabling us to learn a visual 3D object detector with massive 2D (pseudo) labels, even without 3D annotations. To break through the information bottleneck of 2D clues, we explore a new perspective: \textbf{Temporal 2D Supervision}. We propose a temporal 2D transformation to bridge the 3D predictions with temporal 2D labels. Two steps, including homography wrapping and 2D box deduction, are taken to transform the 3D predictions into 2D ones for supervision. Experiments conducted on the nuScenes dataset show strong results (nearly 90\% of its fully-supervised performance) with only 25\% 3D annotations. We hope our findings can provide new insights for using a large number of 2D annotations for 3D perception.

1. Introduction

As a fundamental topic in 3D perception scenes, 3D object detection aims to detect a set of objects in 3D space with 3D-oriented boxes and corresponding categories. Due to the application of deep learning, both LiDAR-based and camera-based frameworks have achieved remarkable progress and superior performance. However, launching such cutting edges is ignited by large-scale data and precise 3D labels. The process of creating an application-level dataset undergoes extremely expensive resources.

To tackle this issue, the weakly supervised learning framework is an economic way to learn a 3D detector. Recent works focus on designing more economical labels to train models instead of expensive 3D labels. Most efforts focus on point-cloud scenes which leverage 2D bounding boxes \cite{29} or reference BEV points \cite{20} to supervise a model for generating coarse proposals. Then a point segmentation model is further employed to deduce the foreground points cloud for the subsequent refinement of 3D bounding boxes. These methods need complex multi-stage pipelines. Besides, WeakM3D \cite{21} explores training the camera-only 3D detector with 2D labels and point cloud proposals. These efforts dramatically reduce the annotation cost, making it easy to scale the data.

However, the above approaches still expose several downsides: 1) both sensor data \textit{i.e.}, LiDAR point cloud, and image, are required for learning a detector of a single-modal; 2) the training pipelines are sophisticated since they usually train with multiple stages and require several independent algorithms (\textit{e.g.}, RANSAC \cite{6} to remove ground parts, and extra point segmentation model, \textit{etc.}). To this end, one question may arise: \textit{could we leverage massive 2D labels (\textit{i.e.}, 2D boxes and category labels) to train a 3D camera-based detector in an end-to-end way without involving point clouds?}

Training a 3D detector with 2D annotations is challenging yet promising. Its challenge is reflected by the information bottleneck of 2D clues (\textit{e.g.}, depth, size and rotation). Whilst it is also promising since some precedents provide valuable
insights to deduce 2D clues for learning 3D representation. A classical application is Multi-View Stereo (MVS) [23, 34], which poses multiple spatial or temporal views to reconstruct 3D scenes with only 2D images. Another advanced application is differentiable rendering [3, 11, 19]. It also leverages 2D image supervision of different views to render a single 3D object. Going into one step, [9] extents the differentiable rendering to learn the 3D shapes and layouts of multiple objects with only 2D dense masks. Inspired by those advances in MVS and differentiable rendering, we explore 3D object detection by digesting the 2D supervision at scale.

In this paper, we aim to develop a hybrid training framework (see Fig. 1) that learns a 3D visual detector with massive 2D labels and a few 3D annotations, even without 3D labels. To digest the 2D supervision in the 3D detector, we explore a new perspective to learn 3D representation: temporal 2D supervision. Our hybrid framework can be built upon any 3D visual detectors (e.g., FCOS3D [27]). We first use a homography matrix to wrap the predicted 3D boxes to the adjacent frames. Afterwards, we introduce a non-parametric projection to deduce the warped 3D boxes to 2D ones, which can be directly supervised by temporal 2D (pseudo) labels. With temporal 2D supervision, 3D information (e.g., depth, size, and rotation) can be learned implicitly. With the novel pipeline, we further observe some corner cases introduced by the moving objects and propose some regularization strategies: 1) symmetrical temporal supervision and 2) appropriate temporal intervals.

We conduct comprehensive experiments on nuScenes dataset [2] to show the effectiveness of our method. Experimental result shows that strong performance (nearly 90% of its fully-supervised performance) are achieved with only 25% 3D annotations. It’s worth noting that our method with 100% 2D supervision can produce a promising result (60% of its 3D oracle performance). With our training framework, it is flexible to use any ratio between 2D and 3D labels, enabling control of the annotation cost in practical applications. We hope our methodology and studies can provide new insights for using large-scale 2D labels for 3D detectors.

2. Related Work

3D object detection focus on detecting objects with 3D-oriented boxes in 3D space. With the great progress in deep learning, LiDAR-based and camera-based 3D detectors are continuously evolving. One stream of LiDAR-based works [24, 32, 33, 37] directly processes vanilla point cloud in 3D space and leverages PointNet++-like network to encode features for prediction. Several works [13, 30, 35, 39] focus on splitting point cloud into regular voxels followed by pooling to 2D representation, after which 2D convolution networks are naturally utilized to process feature maps. In addition, recent efforts aim to conduct 3D object detection tasks based on image modals, which is a more cheap scheme. They employ a 2D backbone to extract feature maps and attach a 3D target-specific head to predict objects in 3D space. CenterNet [38] introduces a heatmap-based head to perceive objects and eliminate the NMS procedure. Inspired by dense prediction framework in 2D tasks [16, 25], FCOS3D [27] predicts objects in each location of the multi-level feature map. The great breakthroughs made by the above 3D detector benefit from creating large-scale datasets with precise 3D annotations. The process of collecting such a dataset undergoes time-consuming and expensive resources so we explore leveraging cheaper 2D labels for training in this paper.

Weakly supervised 3D object detection is a promising way to economically train models. It aims to leverage the partial representation of 3D labels e.g., 2D boxes, and BEV points, to learn a 3D detector. Recently, Meng et al. [20] proposes to annotate a small part of BEV points and precise 3D labels in LiDAR point cloud data for training. It designs a two-stage architecture for predicting results so a two-stage training process is also conducted. To eliminate the requirement of 3D labels, FGR [18] proposes to leverage only 2D boxes in the image to project coarse 3D point cloud space. Then a point segmentation model is used to identify foreground points for the subsequent 3D box prediction. The weakly supervision technologies are well explored in LiDAR-based frameworks but rarely involve camera-based ones until the coming of WeakM3D [21]. It similarly employs 2D boxes in image space to crop point cloud proposals which is leverage to build strong constraints for predictions of cameras. While these works provide several advanced methods for cheap training detectors, they still undergo two issues: 1) both modals need not be absent for carrying out supervision; 2) the training pipeline requires multiple stages and complex algorithms. In this paper, we not only utilize temporal 2D labels for training a camera-based detector without LiDAR point cloud but train our model in an end-to-end manner without extra complex algorithms.

Learning 3D tasks with 2D supervision is also a promising topic in many tasks like Multi-View Stereo (MVS), differentiable rendering, and 3D shape and layout perception etc. By structuring projection between 3D and 2D representations, multi-view or temporal 2D cues may be used to supervise the training process for the 3D perception tasks. MVS technology [23, 34] structures a cost volume among multi-view 2D images and further measures similarity for 3D object structure reconstruction. Differentiable rendering [3, 11, 19] focuses on leveraging 2D representations to flow information to 3D visuals for rendering object details. Recent work [9] delves into the 3D shape and layout prediction framework by training its model with multi-view 2D images. It successfully learns dense depth information and layout in 3D space. Motivated by these precedents, we explore training the monocular 3D detector by temporal 2D supervision. To
the current frame image is input to the monocular detector for predicting 3D representation and classification. During training, a controller is utilized to adjust the ratio between strong and weak supervision. The strong supervision part directly adopts precise 3D ground truth for building loss, while the weak part needs to carry out our proposed transformation for generating predicted temporal 2D representations. The transformation is parameter-free, including the homography warping and the 2D box deduction processes. Therefore, ground truth (pseudo) labels can implicitly supervise 3D representations. The supervision of the classification and center-ness branches is conducted in both strong and weak manners. The velocity, direction, and attribute labels are only usable under strong supervision. The best of our knowledge, it is the first effort to train a camera-based detector with only 2D supervision, which is non-trivial to build projection between sparse 3D and 2D representations.

3. Preliminary

Our framework can be built on any vision detector. In this paper, we mainly explore the hybrid framework based on the monocular 3D detector FCOS3D [27].

Before stepping into our method, we first review the pipeline of FCOS3D [27]. It keeps the fully convolutional network for 2D detection tasks and is applied for 3D object detection by extending the 3D target head. As shown in Fig. 2, an image is first processed by a backbone [10] followed by a feature pyramid network (FPN) [15], which extracts multi-scale feature maps with five different resolutions. Each level of FPN feature map predicts the parameters of 3D box and the corresponding class for each location. It tiles dense anchor points on feature maps with multiple scales and designs a center-based strategy to select the positive and negative samples. Then, two parallel branches are attached for classification and regression, respectively. The classification branch predicts the category or background for each location. As for the regression branch, it predicts 3D representation of all positive locations, involving 3D size \((w, l, h)\), offsets \((\Delta x, \Delta y)\) to the projected 3D center, normalized depth \(d\), rotation \(\theta\), direction class \(C_\theta\), and 3D aware center-ness \(c\).

4. Method

Our method is a hybrid framework for training a camera-only 3D detector with massive cheap 2D labels and a few 3D annotations in an end-to-end manner. For 2D supervision, we adopt temporal 2D bounding boxes for training and attempt to endow the model with the ability to predict 7-DoF 3D boxes in 3D space. While for 3D supervision, we adopt the common way in 3D object detectors.

4.1. Motivation

Learning 3D object detectors with 2D supervision is economical yet challenging due to the lack of valid depth, 3D size, or rotation information. Recent works [21, 29] aim to remedy these incomplete information with the aid of point clouds. By fetching the coarse foreground point clouds projected from 2D supervision, the information from the point cloud can be adopted to supervise 3D predictions. Though these approaches are effective, they still require LiDAR data for training.

In this paper, we explore training vision 3D detectors with pure 2D labels. Recent precedents [3, 9, 11, 19, 23, 34] leverage 2D clues to learn a 3D perception task, e.g., 3D shape and layout estimation. By structuring the mapping or constraint conditions between 3D and multi-view 2D representations, the 3D predictions can be projected to 2D space, supervised by 2D labels. Motivated by these priors, we aim to perform 3D object detection by the supervision of multi-view 2D boxes. However, it is difficult to construct...
In this section, we demonstrate the overall learning pipeline of our method supervised by massive temporal 2D labels and a few 3D labels. As shown in Fig. 2, the 3D detector normally predicts 3D boxes. During training, we design a controller to adjust the proportion between 3D labels and temporal 2D labels for supervision. The 3D supervision part inherits the learning approach of FCOS3D. For 2D supervision, we leverage the non-parameter transformation to transform the predicted 3D boxes into temporal 2D space and deduce them into 2D ones (see Sec. 4.3). By directly employing ground truth (pseudo) 2D labels to supervise these transformed 2D boxes, the representations of 3D predictions can be learned implicitly. Our method is trained in an end-to-end manner and does not need any other clues e.g., LiDAR point cloud, to impose complex constraints.

4.3. Temporal 2D Transformation

In this section, we introduce the temporal 2D transformation to transform the predicted 3D boxes into temporal 2D ones. It is a key condition for supervision with temporal 2D ground truths. As shown in Fig. 2, it mainly includes two steps: homography warping and deducing 2D boxes.

Homography warping. Given an input image $I^t_v$ at frame $t$ from one camera view $v \in V^1$, the detector predicts a set of 3D boxes $\{B^t_i\}_{i=1}^N$ in camera coordinate. Each 3D box is represented by 7-DOF parameters $(x_i, y_i, z_i, w_i, h_i, l_i, \theta_i)$, which are center coordinates, size (width, height, and length), and rotation, respectively. We use the homography matrix to warp the predicted box $B^t_i$ of the current frame $t$ to the camera coordinate space of adjacent frame $B^{t+\Delta t}_i$, where $\Delta t$ indicates the temporal interval between two frames. The homography wrapping process can be formulated as:

$$B^{t+\Delta t}_i = H_{t \rightarrow t+\Delta t} \cdot B^t_i,$$

where $H_{t \rightarrow t+\Delta t}$ is the homography matrix of camera coordinate from $t$ to $t + \Delta t$. Such homography transformation may involve two typical cases: the inner view and outer view (see Fig. 3). It decides that we need to build different homography matrices:

$$\begin{cases}
H_{t \rightarrow t+\Delta t} = C^V_{t}(t) \cdot P_t \cdot P^T_{t+\Delta t} \cdot C^V_{t+\Delta t}^T, & \text{if Inner view} \\
C^V_{t}(t) = C^V_{t+\Delta t}(t+\Delta t), & \text{if Outer view}
\end{cases},$$

where $C^V_{t}(t) \in \mathbb{R}^{4 \times 4}$ is the homogeneous matrix from the image view $V(t)$ of camera coordinate to the ego coordinate at time $t$. $P_t \in \mathbb{R}^{4 \times 4}$ indicates the pose from ego to global coordinate. Both $C$ and $P$ include translation and rotation transformations. Therefore, transforming the predicted 3D boxes of current frame $t$ to adjacent frame $t + \Delta t$ is a homography warping procedure.

Deduce 2D box. After warping the 3D predictions into the camera coordinates of the adjacent frame, here we introduce how to project the 3D representations to 2D boxes. The blue region in the bottom left corner of Fig. 2 shows the projection in detail. Inspired by PGD [26], we project the 3D representations to 2D space by extracting the outsourcing box in the same image perspective. Given a warped 3D box $B^{t+\Delta t}_i \in \mathbb{R}^7$ in a view $V(t + \Delta t)$ at time $t + \Delta t$, we first extract the eight corner points $O^t_{pix} \in \mathbb{R}^{8 \times 3}$ in camera coordinate. Then, the corresponding camera’s intrinsic parameter $K(t+\Delta t) \in \mathbb{R}^{4 \times 4}$ is used to map corners to the image plane:

$$O^t_{pix} = K(t + \Delta t) \cdot O^V_{i}(t+\Delta t),$$

where $O^t_{pix} \in \mathbb{R}^{8 \times 2}$ is the pixel coordinate of projected corners. Afterwards, we calculate the leftmost, rightmost, topmost, and downmost corners for constructing 2D box $D_i \in \{x_{tl}, y_{tl}, x_{br}, y_{br}\}$.

The overall temporal 2D transformation is non-parametric. During training, the predicted 3D boxes can be transformed into 2D boxes of arbitrary adjacent frames, i.e. the $\Delta t$ can be arbitrarily positive or negative values.
Figure 4. **Motion ambiguity.** (a) biased supervision appears in moving objects since it suffers from location shift; (b) the supervision for stationary objects is unbiased since it is a homography warping among temporal frames.

### 4.4. Motion ambiguity

In this section, we reveal that temporal 2D supervision is prone to fall into suboptimal training. We traced the problem to a culprit: **motion ambiguity.** Afterwards, we introduce two strategies to regularize it.

**How does motion ambiguity affect training?** We start out analyzing this issue by probing the instances’ motion distribution of nuScenes dataset [2]. We find there are 26% objects in motion state which is actually the culprit. As shown in Fig. 3 (a), it indicates that the GT 2D boxes in temporally adjacent frames can match the objects. However, by reversing them into the current frame $T$, it triggers location and size misalignment with the same object (GT box). The issue is attributed to the motion of objects, making ambiguity to estimate the 3D information [1, 14, 28] with the homography warping transformation. Therefore, it will cause bias supervision with temporal 2D labels.

**Two regularization strategies.** To alleviate this issue, we find that the object’s motion normally equips with spatiotemporal consistency [4, 31], i.e., the moving tendency of location shift and direction are consistent during a continuous period. Based on this insight, we use symmetrical temporal 2D supervision (e.g., $\Delta t = \{-\tau, 0, \tau\}$) for constructing loss. It can effectively offset the bias among the opposite optimizations. As for the 2D labels in the current frame and all stationary objects, such bias issue is not observed (see Fig. 3 (b)). In addition, we empirically find that reasonably large perspective (e.g., $\Delta t$ is set as $\{-3, 0, 3\}$) achieves the best performance in our practice. The possible reason is that short temporal intervals may not introduce enough parallax for depth estimation and even bring about some noise for supervision. The effectiveness of both strategies will be further verified in Tab. 1c.

### 4.5. Hybrid Training

In this section, we introduce how to train the 3D detector in a hybrid manner with any proportion of 2D labels. To achieve it, we need to formulate the label assignment for both 2D and 3D supervision, which is important [7, 25, 36] for dense prediction frameworks like FCOS3D [27].

**Label assignment.** For both 2D and 3D supervision, we use the center distance prior to assign positive and negative samples, similar to FCOS3D. For the 3D part, FCOS3D adopts the projected 2D center of 3D ground truth in the image plane for reference.

For pure 2D supervision, 2D ground-truth (GT) is short of depth information for projection. We employ the center of 2D labels to replace the projected 2D center, which has little impact on performance (see the “w/o 3D LA” in Tab 1b). In addition, the GT center-ness is also derived from the 2D box center in a 2D Gaussian distribution manner. For more label assignment details, please refer to [27].

**Supervision.** The overall loss function is formulated as:

$$L_{hybrid} = \lambda_{3D}L_{3D} + \lambda_{2D}L_{2D},$$

(4)

where the $\lambda_{2D}$ and $\lambda_{3D}$ are the loss weights for the 2D and 3D supervisions, respectively. For the $L_{3D}$, we follow the same loss function as FCOS3D:

$$L_{3D} = \frac{1}{N_{pos}}(\alpha_{cls} L_{cls} + \alpha_{attr} L_{attr}$$

$$+ \alpha_{loc} L_{loc}^3 + \alpha_{dir} L_{dir} + \alpha_{ct} L_{ct}^3),$$

(5)

where the focal loss and binary cross entropy (BCE) loss are used for the classification branch, and center-ness regression denoted as $L_{cls}$, $L_{ct}$, respectively. Softmax classification loss is used for direction and attribution classifications denoted as $L_{dir}$ and $L_{attr}$. Smooth 1L loss is adopted to regress $(\Delta x, \Delta y, w, h, d, v_x, v_y)$.

For 2D supervision, we adopt the temporal 2D labels $D_{t+\Delta t}$ to supervise the transformed 2D predictions $\hat{D}_{t+\Delta t}$:

$$L_{2D} = \frac{1}{N_{pos}}(\beta_{cls} L_{cls} + \beta_{loc} L_{loc}^2 + \beta_{ct} L_{ct}^2),$$

(6)

where $N_{pos}$ is the number of positive samples. $L_{cls}$, $L_{ct}^2$ are the classification and center-ness losses, respectively. $\beta_{cls}$, $\beta_{loc}$ and $\beta_{ct}$ are the corresponding loss weights. $L_{loc}^2$ is the localization loss and we adopt the GIoU [22] for supervision:

$$L_{loc}^2 = \sum_{i=0}^{N} \sum_{j=\Delta t}^\tau \text{GIoU}(\hat{D}_i^j, D_i^j),$$

(7)

where $\hat{D}_i^j$ is the transformed 2D prediction and $D_i^j$ is temporal 2D label. $\Delta t \in \{-\tau, \ldots, 0, \ldots, \tau\}$ is a temporal series. $N$ is the total number of positive predictions. $\tau$ can be instanced to arbitrarily temporal 2D labels.
5. Experiments

In this section, we conduct our experimental analyses on nuScenes dataset [2]. It starts with the ablation studies on dropping 3D supervisions and adding temporal 2D supervisions step by step in Sec. 5.2. Next, we report the performance on hybrid training between 3D and 2D supervision in Sec. 5.3. Finally, we delve into more interesting results and discussions in Sec. 5.4.

5.1. Experimental Setup

Datasets and Metrics. We conduct the exploration study on nuScenes dataset [2]. It is a large-scale dataset for automatic driving scenes including 700/150/150 scenes for training, validation, and testing, respectively. Each scene is a continuous 20s video containing 40 keyframes with precise 3D annotations. It collects multimodal data from LiDAR, camera, and radar, while we only use the camera data. The official provides several evaluation metrics including mean Average Precision (mAP), mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error (mAOE), mean Average Velocity Error (mAVE), mean Average Attribute Error (mAAE), and nuScenes Detection Score (NDS). For the full 2D supervision setting, we only report mAP, mATE, and mASE. For the joint training setting, we report all evaluation metrics.

Implementation Details. If not specified, we adopt ResNet50 [10] initialized from ImageNet [12] as the backbone for all experiments. Other networks like FPN [15] and head are randomly initialized and trained from scratch in an end-to-end manner. We train the model with SGD optimizer with batch size 16 on 8 GTX 2080Ti GPUs. We set the learning rate to 0.002 and use gradient clip and warm-up strategy for training. The warm-up iterations are set to 500 with the 0.33 ratio. We train the model with 12 epochs for ablation study, while we report the main results with extra 12 epochs of fine-tuning. During training, we use the 2D box projected from the 3D ground truth as our temporal supervision.

5.2. Ablation Study

We first conduct an incremental experiment for adding the 2D temporal supervision on 3D supervised baseline (oracle). Afterwards, we carry out ablation studies on dropping 3D supervisions and adding temporal 2D supervisions step by step. Finally, a simulation experiment is designed to remove the ambiguity of 2D labels achieved from human annotation or projected from 3D annotations.

Ablation 1: strong oracle. We start out reporting the full 3D supervised baseline for comparison in Tab. 1a. It achieves 25.5% mAP and 33.9% NDS performance. Based on the baseline, we add the 2D supervision for the current frame, which boosts the performance by slightly 0.4% mAP and 0.5% NDS. The gains are the same as PGD [26] but we only leverage 2D labels to supervise the projected 2D predictions instead of adding an extra 2D prediction head. By projecting with GT depth for predictions, the performance keeps comparable. When temporal 2D labels participate in supervision, the performance drops slightly, which may be attributed to the biased supervision. Large temporal interval can offset the bias, as maintained in Sec. 4.5.

Ablation 2: removing 3D supervision. To train the model with only 2D labels, we start out using the 2D center for label assignment mentioned in Sec. 4.5 instead of the 3D aware center. Tab. 1b shows slight performance degradation. Next, we drop 3D size and rotation supervision, which largely affects the performance of mASE and mAOE. Then,
we further remove the supervision of offset, it barely affects all performance metrics. Specially, the absence of the above 3D supervision cues only bring about a slightly negative impact on mAP and mATE metrics. The phenomenon largely depends on the definition of mAP (i.e., it measures the center distance to calculate average precision), while mATE metric is also measured by center distance error. Finally, we remove the depth supervision, which significantly damages all performance metrics (i.e., 2.6% mAP, 1.149 mATE, and 0.856 mASE). The result makes sense since 2D labels of single frame will suffer from serious depth information deficiency. It poses a challenge for perceiving the 3D clues.

Ablation 3: effectiveness of the temporal 2D supervision. After hitting rock bottom in performance, we start studying the effectiveness of adding temporal 2D supervision to estimate the 3D information again. The results reported in Tab. 1c can be combined into multiple groups for exploring several interesting findings. By comparing A vs. B (B, C, D), temporal 2D supervision significantly improves nearly 10% mAP, while mATE and mASE achieve over 15% and 30% gains, respectively. It reflects that using temporal 2D labels by our method is effective to perceive depth information. By comparing (B, C, D) vs. (E, F, G), it reveals that symmetrical supervisions outperform unilateral supervision, which agrees with our analysis in Sec. 4.5. We further compare performance inside (E, F, G, H), which reflects that the appropriately large temporal interval can constantly boost performance. Once the temporal interval is set too big, the performance begins to drop. The phenomenon also agrees with our analysis in Sec. 4.5. As for other temporal 2D supervision manners, they are the sub-optimal trade-off schemes between optimal temporal interval and symmetrical supervision. Therefore, we use 2D temporal supervision with [-3, 0, 3] intervals as default setting in subsequent exploration.

Ablation 4: is the 3D prior affect the performance? We further study the 2D labels projected from 3D ground truth whether triggering 3D information leakage for training. To do this, we employ a recent advanced 2D detector [8] pre-trained on COCO [17] to predict the 2D boxes of the overall train set. Then, we simply utilize the greedy Intersection-over-Union (IoU) searching strategy to perform the matching between predicted 2D boxes and projected 2D ground truths. It reveals that the size and location shifts are:

| shifting scale | ≤ 0.05 | ≤ 0.10 | ≤ 0.15 |
|----------------|--------|--------|--------|
| ratio          | 18%    | 61%    | 19%    |

We note that the shifting scale of 98% of predicted 2D boxes is smaller than 0.15. Once given the ground truth 2D annotations, the shifting scale will be smaller than 0.15 too. Based
on the shifting estimation, we study the effect of 3D prior. Tab. 1d reports the performance by randomly jittering the scale of projected 2D GTs. It shows that removing the 3D prior rarely affects the performance, revealing the projected 2D labels can not trigger the 3D information leakage.

5.3. Hybrid Training

Given partial 2D supervision and 3D labels, we study the performance of our hybrid training framework. We conduct extensive experiments on the training set of nuScenes dataset and ablate the ratio between 3D and 2D labels.

Random split on images and instances. We randomly split the dataset based on images and instances before experiments. As shown in Fig. 5, it shows that our hybrid training can constantly improve the mAP and NDS performance by replacing the 3D supervision with temporal 2D ones. Only given 5% 3D labels, our method can boost the mAP and NDS performance nearly twice. While providing 25% 3D labels, our method achieves 87.5% mAP and 88.9% NDS performances of the full 3D supervision. In addition, it reveals that the random instance split manner performs better than the image split one. It introduces a practical insight to only annotate partial 3D GTs. Workers can select easier instances to annotate, enabling them to reduce massive costs.

Could the split priors on motion state, distance, and size work better? We further study employing priors on the instance division. Fig. 5 reveals that only providing 3D labels for moving instances can achieve improvement but underperform the random split manner. As for the distance and size split manners, only the large size prior shows better performance compared with the random image manner. Unexpectedly, other split priors like remote distance and medium size bring about performance degradation. These interesting findings reveal that random annotation works well, which may introduce a better distribution between the 3D and 2D supervision for optimization.

Main Results. Tab. 2 report the performances on large backbone ResNet101 with DCN [5]. It demonstrates that our hybrid training framework can improve all evaluation metrics. When training with both 12 epochs and continually fine-tuning (the same as FCOS3D [27]), our method can achieve promising performance with small ratio of 3D labels.

5.4. More Interesting Results and Discussions

Performance on different categories and distances. Tab 3 shows traffic cone and barrier classes achieve unexpected performances (i.e., only 7.0% and 2.5% mAP smaller than full 3D supervision) with full 2D supervision. Since these two classes are stationary objects, the temporal 2D supervision will not trigger the bias loss. However, the pedestrian and bus classes perform badly compared with full 3D supervision. We deem it attributes to larger or smaller 3D sizes. In the motion state, the minimum size class, i.e., pedestrians, will trigger a large location shift deviated from the homography warping transformation. As for the large size class like buses and trucks, they are prone to encounter mutation of size due to the drastic visual surface changing. Through the fair comparison between the random sample (within image) 3D supervision and the fully 2D one, we find that the former predicts the moving objects well and the latter perceives stationary objects better. See Tab. 4, we also report the performance on different distances. It indicates that temporal 2D supervision predicts nearby objects well, which is lower than full 3D supervision by 9.4% mAP. While the distant objects are poorly estimated mainly due to the lack of depth.

Discussion and limitation. Our method shows promising potential for training 3D visual detectors with large-scale cheap 2D labels. The overall pipeline is simple and does not need the point cloud data. In addition, the full 2D supervision manner will liberate the annotation procedure since it does not require any point clouds for reference.

One nonnegligible limitation of our work is the motion ambiguity under the homography warping procedure. We introduce symmetry supervision to alleviate the issue, but it is far from enough. It poses a significantly valuable but challenging topic for future work.

6. Conclusion

In this paper, we present the visual 3D detector supervised by massive 2D labels and a few 3D ones. With the temporal 2D supervision in our training pipeline, any 3D detectors can effectively estimate the 3D information with 2D annotations. The comprehensive experiments show promising results. We hope our exploration can provide new insights for using a large number of 2D annotations for 3D perception.
References

[1] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Multi-view depth estimation by fusing single-view depth probability with multi-view geometry. In CVPR, 2022. 5

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In CVPR, 2020. 2, 5, 6

[3] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In NeurIPS, 2019. 2, 3

[4] Ting-Wu Chin, Ruizhou Ding, and Diana Marculescu. Adascale: Towards real-time video object detection using adaptive scaling. Proceedings of Machine Learning and Systems, 2019. 5

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In ICCV, 2017. 8

[6] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, pages 381–395, 1981. 1

[7] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. Ota: Optimal transport assignment for object detection. In CVPR, 2021. 5

[8] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430, 2021. 7

[9] Georgia Gkioxari, Nikhila Ravi, and Justin Johnson. Learning 3d object shape and layout without 3d supervision. In CVPR, 2022. 2, 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016. 3, 6

[11] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In CVPR, 2018. 2, 3

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 2017. 6

[13] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In CVPR, 2019. 2

[14] Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with dynamic temporal stereo. arXiv preprint arXiv:2209.10248, 2022. 5

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In CVPR, 2017. 3, 6

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In ICCV, 2017. 2

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In CVPR, 2014. 7

[18] Haizhuang Liu, Huimin Ma, Yilin Wang, Bochao Zou, Tianyu Hu, Rongquan Wang, and Jiashen Chen. Eliminating spatial ambiguity for weakly supervised 3d object detection without spatial labels. In ACM MM, 2022. 2

[19] Shichen Liu, Weikai Chen, Tianye Li, and Hao Li. Soft rasterizer: Differentiable rendering for unsupervised single-view mesh reconstruction. arXiv preprint arXiv:1901.05567, 2019. 2, 3

[20] Qinghao Meng, Wenguang Wang, Tianfei Zhou, Jianbing Shen, Luc Van Gool, and Dengxin Dai. Weakly supervised 3d object detection from lidar point cloud. In ECCV, 2020. 1, 2

[21] Liang Peng, Senbo Yan, Boxi Wu, Zheng Yang, Xiaofei He, and Deng Cai. Weak3d: Towards weakly supervised monocular 3d object detection. In ICLR, 2022. 1, 2, 3

[22] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In CVPR, 2019. 5

[23] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV, 2002. 2, 3

[24] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In CVPR, 2019. 2

[25] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In ICCV, 2019. 2, 5

[26] Tai Wang, ZHU Xinge, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In CoRL, 2022. 4, 6

[27] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In ICCV, 2021. 2, 3, 5, 7, 8

[28] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunistic self-supervised multi-frame monocular depth. In CVPR, 2021. 5

[29] Yi Wei, Shang Su, Jiwen Lu, and Jie Zhou. FGR: frustum-aware geometric reasoning for weakly supervised 3d vehicle detection. In ICRA, 2021. 1, 3

[30] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded 3d object detection and tracking. In CVPR, 2021. 2

[31] Jinrong Yang, Songtao Liu, Zeming Li, Xiaoping Li, and Jian Sun. Real-time object detection for streaming perception. In CVPR, 2022. 5

[32] Jinrong Yang, Lin Song, Songtao Liu, Zeming Li, Xiaoping Li, Hongbin Sun, Jian Sun, and Nanning Zheng. Dbq-ssd: Dynamic ball query for efficient 3d object detection. arXiv preprint arXiv:2207.10999, 2022. 2

[33] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In CVPR, 2020. 2

[34] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In ECCV, 2018. 2, 3

[35] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In CVPR, 2021. 2
[36] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In CVPR, 2020.
[37] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In CVPR, 2022.
[38] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. arXiv preprint arXiv:1904.07850, 2019.
[39] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In CVPR, 2018.