Incident Detection in Traffic Video

Project-II (EC47004) report submitted to

Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of

Bachelor of Technology

in

Electronics and Electrical Communication Engineering

by

Ashutosh Naik

(20EC39049)

Under the supervision of

Prof. Pabitra Mitra

Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

Spring Semester, 2024

April 30, 2024

DECLARATION

I certify that

(a) The work contained in this report has been done by me under the guidance of my supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: April 30, 2024 (Ashutosh Naik)

Place: Kharagpur (20EC39049)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**Incident Detection in Traffic Video**" submitted by **Ashutosh Naik** (Roll No. 20EC39049) to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Bachelor of Technology in Electronics and Electrical Communication Engineering is a record of bona fide work carried out by him under my supervision and guidance during Spring Semester, 2024.

Pabitra Mitra

Prof. Pabitra Mitra
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur - 721302, India

Date: April 30, 2024

Place: Kharagpur

Abstract

Name of the student: **Ashutosh Naik** Roll No: **20EC39049**

Degree for which submitted: **Bachelor of Technology**

Department: **Department of Computer Science and Engineering**

Thesis title: **Incident Detection in Traffic Video**

Thesis supervisor: **Prof. Pabitra Mitra**

Month and year of thesis submission: **April 30, 2024**

The paper tackles the issue of ambiguity in ground-truth annotations of 3D bounding boxes, which can arise due to occlusions, missing data, or manual annotation errors, and can negatively impact the training of deep 3D object detectors. To address this, it formulates the label uncertainty problem as the diversity of potential bounding boxes for objects. It proposes GLENet, a generative framework adapted from conditional variational autoencoders, to model the one-to-many relationship between 3D objects and their potential ground-truth bounding boxes using latent variables. GLENet's generated label uncertainty is a plug-and-play module that can be integrated into existing 3D detectors to build probabilistic detectors and supervise localization uncertainty learning. The paper also proposes an uncertainty-aware quality estimator architecture to guide the training of the IoU-branch with predicted localization uncertainty. When incorporated into various 3D detectors, the proposed methods demonstrate significant performance gains on KITTI and Waymo datasets, with GLENet-VR outperforming all published LiDAR-based approaches on the challenging KITTI test set.

Acknowledgements

I would like to thank my Thesis supervisor Prof. Pabitra Mitra for directing, guiding, and supporting me throughout the project.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

In the realm of urban mobility and autonomous transportation, the effective detection of incidents in traffic videos stands as a pivotal challenge with far-reaching implications. As road networks continue to evolve and the deployment of autonomous vehicles becomes more commonplace, ensuring the swift and accurate identification of anomalous events, such as accidents or hazards, becomes paramount for enhancing overall traffic safety. Incident detection serves as the linchpin for proactive responses, enabling timely interventions to prevent collisions, protect pedestrians, and optimize traffic flow. Beyond its fundamental role in safeguarding lives and infrastructure, robust incident detection holds the key to unlocking the full potential of smart transportation systems, paving the way for efficient traffic management and the seamless integration of autonomous vehicles into our urban landscapes.

1.2 Motivation

The motivation behind incident detection in traffic videos is rooted in its profound impact on public safety and the well-being of individuals. Swift and accurate identification of incidents, such as accidents or hazards, plays a pivotal role in saving lives and preventing injuries. When incidents are promptly recognized, emergency

services can be mobilized efficiently, reducing response times and potentially minimizing the severity of outcomes.

Moreover, incident detection is instrumental in limiting property damage. By quickly identifying and responding to accidents, interventions can be initiated to prevent further destruction and protect valuable assets. This not only safeguards individuals but also contributes to the preservation of infrastructure and resources.

The implications extend beyond immediate response to incidents. In the realm of insurance, incident detection in traffic videos becomes a critical tool for evaluating claims. Insurance companies can use this information to determine fault, assess damages, and decide on appropriate compensation. This not only streamlines the claims process but also aids insurance companies in making informed decisions about their customers, helping to tailor coverage plans and premiums based on individual risk profiles.

In legal contexts, incident detection serves as valuable evidence for police investigations. The captured footage provides a factual and unbiased account of events, aiding law enforcement in understanding the sequence of incidents and establishing responsibility. This not only expedites legal proceedings but also ensures a fair and accurate representation of events.

Overall, the motivation for incident detection in traffic videos is deeply embedded in its potential to save lives, reduce injuries, limit property damage, facilitate insurance processes, and contribute to the effective resolution of legal cases. As technology continues to advance, harnessing the power of incident detection becomes increasingly vital for creating safer, more secure, and well-managed transportation ecosystems.

1.3 Objective

The core objective is to train the GLENet architecture model for accurate accident detection in traffic videos, providing detailed incident descriptions. The study also aims to enhance the model's predictive capabilities, offering relative confidence

levels for each detection. This research contributes to the development of a transparent and effective incident detection system, crucial for advancing traffic safety and autonomous vehicle deployment.

Chapter 2

Literature Review

2.1 LiDAR-based 3D Object Detection

Here is the paraphrased version including all the references mentioned in the original passage:

Existing 3D object detectors can be broadly classified into two categories: single-stage and two-stage. For single-stage detectors, Zhou and Tuzel (2018) proposed converting raw point clouds to regular volumetric representations and adopted voxel-based feature encoding. Yan et al. (2018b) presented a more efficient sparse convolution approach. Lang et al. (2019) converted point clouds to sparse fake images using pillars. Shi and Rajkumar (2020a) aggregated point information via a graph structure. He et al. (2020) introduced point segmentation and center estimation as auxiliary tasks during training to enhance model capacity. Zheng et al. (2021a) constructed an SSFA module for robust feature extraction and a multi-task head for confidence rectification, and proposed DI-NMS for post-processing.

For two-stage detectors, Shi et al. (2020b) exploited a voxel-based network to learn the additional spatial relationship between intra-object parts under the supervision of 3D box annotations. Shi et al. (2019) proposed directly generating 3D proposals from raw point clouds in a bottom-up manner, using semantic segmentation to validate points and regress detection boxes. The follow-up work by Yang et al. (2019) further proposed PointsPool to convert sparse proposal features to compact

representations and used spherical anchors to generate accurate proposals. Shi et al. (2020a) utilized both point-based and voxel-based methods to fuse multi-scale voxel and point features. Deng et al. (2021) proposed voxel RoI pooling to extract RoI features from coarse voxels.

To address boundary ambiguity problems in 3D object detection caused by occlusion and signal miss, some studies like SPG (Xu et al., 2021), Yan et al. (2021), and Najibi et al. (2020) have explored using point cloud completion methods to restore the full shape of objects and improve detection performance. However, generating complete and precise shapes from incomplete point clouds remains a non-trivial task.

2.2 Probabilistic 3D Object Detector

There are two main categories of uncertainty that can affect deep learning predictions. The first is aleatoric uncertainty, which stems from inherent noise or randomness present in the observational data itself. This type of uncertainty is unavoidable and cannot be eliminated. The second category is epistemic or model uncertainty, which arises due to incomplete training data or lack of knowledge. This type of uncertainty can potentially be reduced by providing more comprehensive training data to the model.

Most current state-of-the-art 2D (e.g., Liu et al., 2016; Tan et al., 2020; Carion et al., 2020) and 3D (e.g., Shi et al., 2020b) object detectors generate deterministic bounding box predictions along with a confidence score. While the confidence score reflects the model's certainty about the existence and semantic classification of the object, it does not adequately capture the uncertainty associated with the predicted localization of the bounding box.

In contrast to these deterministic approaches, probabilistic object detectors (such as those proposed by He et al., 2019; Harakeh et al., 2020; Varamesh and Tuytelaars, 2020) aim to estimate the probability distribution of the predicted bounding boxes, rather than treating them as fixed, deterministic values. For instance, methods like He et al. (2019) and Choi et al. (2019) model the predicted bounding boxes as

Gaussian distributions, where the variance can indicate the level of uncertainty in the localization.

However, most existing probabilistic detectors assume that the ground-truth bounding box annotations are deterministic and can be modeled as Dirac delta distributions, failing to account for the inherent ambiguity present in these annotations. As a result, the localization uncertainty is learned in an unsupervised manner, which may lead to suboptimal localization precision and unstable training behavior

2.3 Label Uncertainty Estimation

Label noise/uncertainty in real-world datasets can severely impact supervised learning, as neural networks can overfit even random noise (Zhang et al., 2021). Previous approaches either remove misclassified samples as uncertain (Delany et al., 2012), use soft voting to estimate noise levels (Garcia et al., 2015; Luengo et al., 2018), or model the joint distribution of noisy and true labels (Northcutt et al., 2021) - but focus mainly on image classification.

For bounding box uncertainty, Meyer and Thakurdesai (2020) modeled it as IoU between the box and LiDAR convex hull, but this non-learning approach has limited capacity and only produces whole-box uncertainty. Wang et al. (2020) used a Bayesian method quantifying point cloud matching to the box via a Gaussian Mixture Model, but violates the point cloud conditional independence assumption.

In contrast, this work formulates label uncertainty as the diversity of plausible bounding boxes. For sparse objects, GLENet outputs different plausible boxes to estimate high uncertainty, regardless of point matching to the given label. Unlike Wang et al.'s (2020) Bayesian paradigm estimating annotated box correctness, this models uncertainty as the diversity of potential boxes predicted by GLENet.

2.4 Conditional Variational Auto-Encoder

The variational autoencoder (VAE) (Kingma and Welling, 2014) has been widely utilized for image and shape generation tasks (Yan et al., 2016; Nash and Williams, 2017). It transforms input samples into a distribution, from which latent variables can be sampled and passed to a decoder network to generate diverse outputs. Sohn et al. (2015) introduced the conditional variational autoencoder (CVAE), which extends the traditional VAE by incorporating an additional condition during the generative process. The CVAE consists of an encoder, decoder, and an extra input like a label or structured information relevant to the generation task. This auxiliary condition allows CVAE to generate more targeted and controlled samples compared to the unsupervised VAE.

In natural language processing, VAE has been widely applied to text generation tasks such as dialogue response (Zhao et al., 2017), machine translation (Zhang et al., 2016), story generation (Wang and Wan, 2019), and poem composition (Li et al., 2018). VAE and CVAE have also found applications in computer vision tasks like image generation (Yan et al., 2016), human pose estimation (Sharma et al., 2019), medical image segmentation (Painchaud et al., 2020), salient object detection (Li et al., 2019; Zhang et al., 2020), and modeling human motion dynamics (Yan et al., 2018a). Recently, these algorithms have been extensively applied to 3D point cloud applications such as generating grasp poses (Mousavian et al., 2019) and instance segmentation (Yi et al., 2019).

Inspired by CVAE's ability to generate diverse reasonable responses in dialogue systems, this work proposes GLENet, adapted from CVAE, to capture the one-to-many relationship between objects with incomplete point clouds and their potentially plausible ground-truth bounding boxes. To the best of the authors' knowledge, this is the first work to employ CVAE in 3D object detection to model label uncertainty.

Chapter 3

Methodology

3.1 Proposed Label Uncertainty Estimation

The ambiguity of annotated ground-truth labels is a prevalent issue in 3D object detection scenarios that adversely affects deep model learning, but has been largely unaddressed or ignored by previous works. To tackle this, we propose GLENet, a generic deep learning framework that models the one-to-many relationship between point cloud objects and potentially plausible bounding box labels to generate label uncertainty. GLENet outputs multiple bounding box predictions for a single object, and the variance among these predictions is computed as the label uncertainty. This label uncertainty is then incorporated as an auxiliary regression objective to enhance the performance of the downstream 3D object detection task, providing a principled approach to account for label ambiguity and improve the robustness and accuracy of 3D object detectors.

3.1.1 Problem formulation

Let $\mathcal{C} = \{c_i\}_{i=1}^{n}$ be a set of n observed LiDAR points belonging to an object, where $c_i \in \mathbb{R}^3$ is a 3D point represented with spatial coordinates. Let X be the annotated ground-truth bounding box of $\mathcal{C}$ parameterized by the center location (c_x, c_y, c_z), the size (l, w, h), and the orientation r, i.e., $X = [c_x, c_y, c_z, w, l, h, r] \in \mathbb{R}^7$. We formulate

the uncertainty of the annotated ground-truth label of an object as the diversity of potentially plausible bounding boxes of the object, which could be quantitatively measured with the variance of the distribution of the potential bounding boxes. First, we model the distribution of these potential boxes conditioned on point cloud $\mathcal{C}$, denoted as $p(X|\mathcal{C})$. Specifically, based on the Bayes theorem, we introduce an intermediate variable z to write the conditional distribution as :

$$p(X|C) = \int_z p(X|z, C)p(z|C), dz$$

Then, with $p(X|z, C)$ and $p(z|C)$ known, we can adopt a Monte Carlo method to get multiple bounding box predictions by sampling z multiple times and approximate the variance of $p(X|C)$ with that of the sampled predictions. In the following, we will introduce our learning-based framework named GLENet to realize the estimation process.

3.1.2 Inference Process of GLENet

GLENet has parameters θ to predict $p(z|C)$ and $p(X|z, C)$. Specifically, it assumes that the prior distribution of a latent variable z, denoted as $p(z|C)$, follows a multivariate Gaussian distribution parameterized by (μ_z, σ_z). The model consists of the following: A prior network, composed of PointNet and additional MLP layers, takes the input point cloud C and predicts the values of (μ_z, σ_z). A context encoder embeds the input point cloud C into a high-dimensional geometric feature representation f_C. The latent variable z is sampled from the multivariate Gaussian distribution $\mathcal{N}(\mu_z, \sigma_z^2)$. The sampled z is concatenated with the geometric feature representation f_C and fed into a prediction network composed of MLPs to regress the bounding box distribution $p(X|z, C)$, which encodes the localization, dimension, and orientation of the bounding box.

As empirically observed in various related domains, it can be challenging to effectively utilize latent variables when the prediction network can generate plausible outputs solely based on the sufficiently expressive features of the condition C. Consequently, to prevent posterior collapse, GLENet employs a simplified PointNet architecture as the backbone of the context encoder. Additionally, $p_\theta(z|C)$, $p_\theta(X|z, C)$,

and $p_\theta(X|C)$ are used to denote the predictions of $p(z|C)$, $p(X|z,C)$, and $p(X|C)$ by GLENet, respectively.

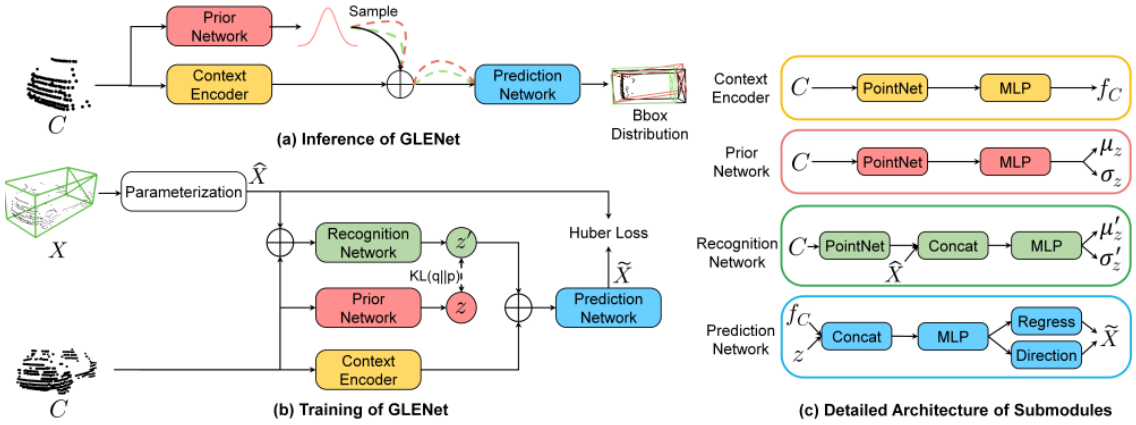


FIGURE 3.1: Overall wrorkflow of GLENet

3.1.3 Training Process

3.1.3.1 Recognition Network

Given an input point cloud C and its annotated ground-truth bounding box X, the GLENet model assumes the existence of a true posterior distribution $q(z|X,C)$ of the latent variable z. During training, a recognition network parametrized by ϕ is employed to learn an auxiliary posterior distribution $q_\phi(z'|X,C)$, which follows a Gaussian distribution $\mathcal{N}(\mu_{z'}, \sigma^2_{z'})$. This recognition network aims to regularize the prior distribution $p_\theta(z|C)$ predicted by the prior network, ensuring that $p_\theta(z|C)$ is close to $q_\phi(z'|X,C)$. The recognition network shares the same learning architecture as the prior network, generating point cloud embeddings that are concatenated with the ground-truth bounding box information X. This concatenated representation is then fed into subsequent MLP layers to learn the parameters $\mu_{z'}$ and $\sigma_{z'}$ of the auxiliary posterior distribution $q_\phi(z'|X,C)$.

Encoding the information X into offsets relative to predefined anchors to facilitate the learning and normalizing :

$$t_{cx} = \frac{c_{gt_x}}{d_a}, t_{cy} = \frac{c_{gt_y}}{d_a}, t_{cz} = \frac{c_{gt_z}}{h_a}$$

$$t_w = \log\left(\frac{w_{gt}}{w_a}\right), t_l = \log\left(\frac{l_{gt}}{l_a}\right), t_h = \log\left(\frac{h_{gt}}{h_a}\right)$$

$$t_r = \sin(r_{gt})$$

(w^a, l^a, h^a) is the size of the predefined anchor located in the center of the point cloud.

$$d_a = \sqrt{(l_a)^2 + (w_a)^2}$$

We also take $\cos(r)$ as the additional input of the recognition network to handle the issue of angle periodicity, r being the rotation angle.

Optimizing the GLENet by maximizing the variational lower bound of the conditional log-likelihood $p_\theta(X|C)$

$$\log p_\theta(X|C) \geq \mathbb{E}_{q_\phi(z'|X,C)}[\log p_\theta(X|z', C)] - KL(q_\phi(z'|X, C)||p_\theta(z|C))$$

The reconstruction loss for the bounding box can be given as:

$$L_{rec} = L_{rec}^{reg} + \lambda L_{rec}^{dir}$$

where L_{rec}^{reg} denotes the Huber loss imposed on the prediction and encoded regression targets and L_{rec}^{dir} denotes the binary cross-entropy loss used for direction classification.

Since $p_\theta(z|C)$ and $q_\phi(z'|X, C)$ are re-parameterized as $\mathcal{N}(\mu_z, \sigma_z^2)$ and $\mathcal{N}(\mu_z', \sigma_z'^2)$ through the prior network and the recognition network, the regularization loss can be given as:

$$L_{KL}(q_\phi(z'|X, C)\|p_\theta(z|C)) = \log\frac{\sigma_z'}{\sigma_z} + \frac{\sigma_z^2}{2\sigma_z'^2} + \frac{(\mu_z - \mu_z')^2}{2\sigma_z'^2}$$

Overall objective function :

$$L = L_{rec} + \gamma L_{KL}$$

Empirically, $\gamma = 1$

3.2 Probabilistic 3D Detectors with Label Uncertainty

We can enforce the detection head to estimate a probability distribution over bounding boxes, $P_\Theta(y)$ instead of a deterministic bounding box location:

$$P_\Theta(y) = \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} e^{-\frac{(y-\hat{y})^2}{2\hat{\sigma}^2}}$$

where Θ represents weights of the detector, $\hat{y}$is the predicted bounding box location, and $\hat{\sigma}$ is the predicted localization variance.

Also, assuming the ground-truth bounding box as a a Gaussian distribution $P_D(y)$ with variance σ^2, the value is estimated by GLENet:

$$P_D(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-y_g)^2}{2\sigma^2}}$$

Incorporating the uncertainty in KL loss between the distribution of prediction and ground-truth in the detection head: Here's the LaTeX code for the given expression:

$$L_{reg} = D_{KL}(P_D(y)||P_\Theta(y)) = \log\frac{\hat{\sigma}}{\sigma} + \frac{\sigma^2}{2\hat{\sigma}^2} + \frac{(y_g-\hat{y})^2}{2\hat{\sigma}^2}$$

If our ground truth would have been direc delta,

$$L_{reg}^{prob} \propto \frac{1}{2}\log(\hat{\sigma}^2) + \frac{(y_g-\hat{y})^2}{2\hat{\sigma}^2}$$

$$\frac{\partial L_{reg}^{prob}}{\partial\hat{\sigma}} = \frac{1}{\hat{\sigma}} - \frac{(y_g-\hat{y})^2}{\hat{\sigma}^3}$$

Now as $|y_g - \hat{y}| \to 0$

$$\frac{\partial L_{reg}^{prob}}{\partial\hat{\sigma}} \to \frac{1}{\hat{\sigma}}$$

the gradient explodes when $\hat{\sigma} \to 0$, but the prediction is optimal only when $\hat{\sigma} = 0$ and $|y_g - \hat{y}| = 0$. So, the gradient explosion may result in erratic training and sub-optimal localization precision.

In contrast, the Gaussian model predicts:

$$\frac{\partial L_{reg}}{\partial \hat{\sigma}} = \frac{1}{\hat{\sigma}} - \frac{\sigma^2}{\hat{\sigma}^3} - \frac{(y_g - \hat{y})^2}{\hat{\sigma}^3}$$

and

$$\frac{\partial L_{reg}}{\partial \hat{y}} = \frac{\hat{y} - y_g}{\hat{\sigma}^2}$$

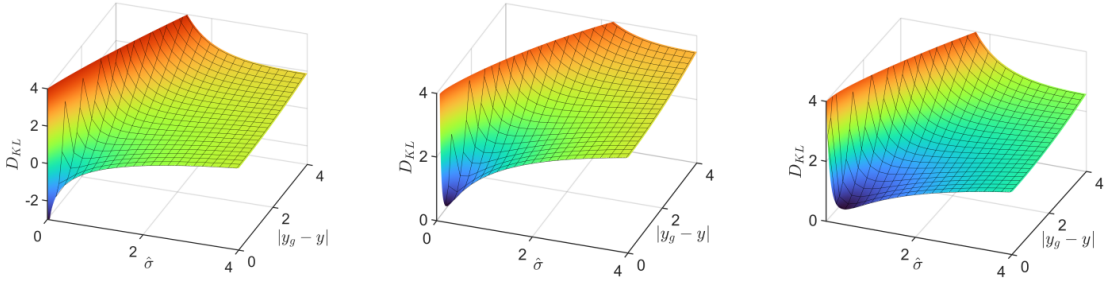
As $|y_g - \hat{y}| \to 0$

$$\frac{\partial L_{reg}}{\partial \hat{\sigma}} \to \frac{1}{\hat{\sigma}} \left(1 - \frac{\sigma^2}{\hat{\sigma}^2}\right)$$

and

$$\frac{\partial L_{reg}}{\partial \hat{y}} \to 0$$

when the predicted distribution reaches the optimal solution that is the distribution of ground-truth $|y_g - \hat{y}| \to 0$ and $\hat{\sigma} \to \sigma$ which avoids gradient explosion.



(a) L_{reg}^{prob} ($\sigma = 0$) (b) L_{reg} ($\sigma = 0.2$) (c) L_{reg} ($\sigma = 0.5$)

FIGURE 3.2: KL divergence as a function of $|y_g - \hat{y}|$, σ and $\hat{\sigma}$

The $L_{prob_{reg}}$ approaches infinitesimal and the gradient explodes as $|y_g - \hat{y}| \to 0$ and $\hat{\sigma} \to 0$. However, when we introduce the estimated label uncertainty and the predicted distribution is equal to the ground-truth distribution, the KL Loss has a determined minimum value of 0.5 and the gradient is smoother.

3.2.1 Uncertainty-aware Quality Estimator

An IoU related confidence score as the sorting criterion is used for Non-Maximum Supression indicating the localization quality. There is a strong correlation between

IoU and uncertainty. However, the estimated uncertainty is 7-dimensional, making it infeasible to directly replace the IoU confidence score with the uncertainty. So, an uncertainty-aware quality estimator (UAQE) is used) that introduces uncertainty information to facilitate the training of the IoU-branch and improve the accuracy of IoU estimation.

FIGURE 3.3: (a)relationship between the actual localization precision (i.e., IoU between predicted and ground-truth bounding box) and the variance (reduced by PCA) predicted by a probabilistic detector. (b) sparse sample - high uncertainty, low localization, while dense sample - low uncertainty

It takes the predicted uncertainty as input and generates a coefficient through two fully connected layers and a Sigmoid activation. This coefficient is then multiplied with the original output of the Intersection over Union (IoU) branch to obtain the final estimation. The UAQE captures the uncertainty in the estimation and adjusts the final output accordingly, resulting in a more accurate estimation of the IoU score.

3.2.2 3D Variance Voting

In probabilistic object detectors, the localization variance learned through the Kullback-Leibler (KL) loss can reflect the uncertainty of the predicted bounding boxes. To leverage this uncertainty information and seek a more precise box representation, we propose a 3D variance voting approach to combine neighboring bounding boxes. At each iteration, the box with the maximum score is selected, and its new location is calculated based on itself and its neighboring boxes. During this merging process, closer neighboring boxes with low variance are assigned higher weights, as

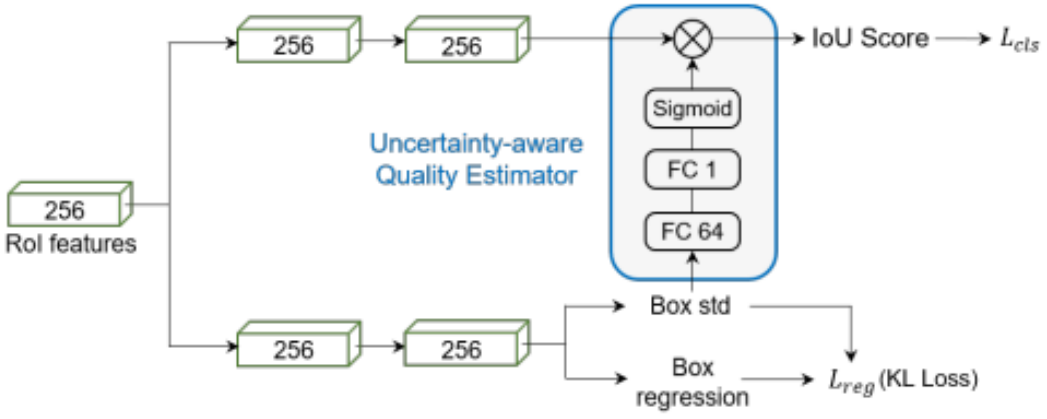


FIGURE 3.4: UAQE module

they are considered more reliable. However, neighboring boxes with a large angle difference from the selected box are excluded from the ensembling of angles, as their orientation information is deemed less relevant. By incorporating the localization uncertainty and spatial proximity of neighboring predictions, this 3D variance voting technique aims to refine the bounding box representation and improve overall detection accuracy.

3.3 Main Function Code

```
from collections import namedtuple

import numpy as np
import torch

from .detectors import build_detector

try:
    import kornia
except:
    pass
    # print('Warning: kornia is not installed. This package is only required by CaDDN')



def build_network(model_cfg, num_class, dataset):
    model = build_detector(
        model_cfg=model_cfg, num_class=num_class, dataset=dataset
```

Algorithm 1: 3D Variance Voting

Data: B is an $N \times 7$ matrix of predicted bounding boxes with parameters $(x, y, z, w, l, h, \theta)$. C is the corresponding variance. S is a set of N corresponding confidence values. σ_t is a tunable hyperparameter.

Result: The final voting results D of selected candidate boxes.

$B = \{b_1, b_2, ..., b_N\}$; and $C = \{c_1, c_2, ..., c_N\}$;
$S = \{s_1, s_2, ..., s_N\}$; and $L = \{1, 2, ..., N\}$;
$D \leftarrow \{\}$;
$iou_{thresh} \leftarrow \mu$;
while $L \neq \emptyset$ **do**
  idx $= \underset{i \in L}{\text{argmax}}\, S$, $b' = b_{idx}$;
  $L' = \{i | i \in L, IoU(b_i, b') > iou_{thresh}\}$;
  $P \leftarrow \{\}$;
  for $i \in L'$ **do**
    $p_i = e^{-(1-IoU(b_i,b))^2/\sigma_t}$;
    if $|tan(b_i^\theta - b'^\theta)| > 1$ **then**
      $p_i^\theta = 0$;
    end
    $P \leftarrow P \bigcup p_i$;
  end
  $b_m = \frac{\sum_{i \in L'} b_i \cdot p_i / c_i}{\sum_{i \in L'} p_i / c_i}$, $p_i \in P, b_i \in B, c_i \in C$;
  $D \leftarrow D \bigcup b_m$;
  $L \leftarrow L - L'$;
end

FIGURE 3.5: Algorithm for 3D variance voting

```
    )
    return model


def load_data_to_gpu(batch_dict):
    for key, val in batch_dict.items():
        if not isinstance(val, np.ndarray):
            continue
        elif key in ['frame_id', 'gt_id', 'metadata', 'calib']:
            continue
        elif key in ['images']:
            batch_dict[key] = kornia.image_to_tensor(val).float().cuda().contiguous()
        elif key in ['image_shape']:
            batch_dict[key] = torch.from_numpy(val).int().cuda()
        else:
            batch_dict[key] = torch.from_numpy(val).float().cuda()
```

```
def model_fn_decorator():
    ModelReturn = namedtuple('ModelReturn', ['loss', 'tb_dict', 'disp_dict'])

    def model_func(model, batch_dict):
        load_data_to_gpu(batch_dict)
        ret_dict, tb_dict, disp_dict = model(batch_dict)

        loss = ret_dict['loss'].mean()
        if hasattr(model, 'update_global_step'):
            model.update_global_step()
        else:
            model.module.update_global_step()

        return ModelReturn(loss, tb_dict, disp_dict)

    return model_func
```

Listing 3.1: Main function code

Chapter 4

Experiment

GLENet was integrated into many 3D object detection frameworks to form probabilistic detectors and evaluated on Waymo Open Dataset (WOD) and KITTI Dataset

4.1 Dataset

KITTI Dataset : 3712 training samples, 3769 validation samples and 7518 testing samples.

Waymo Open Dataset : created a representative training set by randomly selecting 20% of the frames from the original training set, which comprises approximately 32,000 frames. All evaluations were performed on the complete validation set, consisting of around 40,000 frames

4.2 Evaluation Metrics

mean Average Precision (mAP) and mean Average Precision weighted by heading accuracy (mAPH) on WOD. For the GLENet, the negative log-likelihood between the estimated distribution of ground-truth $p_D(X|C)$ given $N(\hat{t}, \sigma^2)$ is calculated due

to the unavailability of the true distribution of a ground-truth bounding box :

$$L_{NLL}(\theta) = -\int p_\theta(X|C) \log p_D(X|C) dX$$

$$\approx -\frac{1}{S} \sum_{i=1}^{S} \log p_D(X_i|C)$$

$$-\frac{1}{S} \sum_{i=1}^{S} \sum_{k \in c_x, c_y, c_z, w, l, h, r} \left(\frac{(t_k^i - \hat{t}_k^i)^2}{2\sigma_k^2} + \frac{1}{2} \log(\sigma_k^2) + \frac{1}{2} \log(2\pi) \right)$$

In the expression, S denotes the number of inferences or sampling iterations, X_i represents the bounding box result from the i-th inference or sample, and $\hat{t}_k^i$ and t_k^i denote the regression targets (ground truth offsets) and the predicted offsets, respectively, for the k-th component of the bounding box. The integral is approximated using the Monte Carlo method by randomly sampling multiple prediction results. Generally, the value of L_{NLL} is small when GLENet outputs reasonable bounding boxes, indicating that it predicts diverse plausible boxes with high variance for incomplete point clouds and consistent, precise boxes with low variance for high-quality point clouds.

4.3 Training Details

As the initial input of GLENet, the point cloud of each object was uniformly preprocessed into 512 points via random subsampling / upsampling. Then coordinates of the center point is subtracted from the point cloud structure to eliminate the local impact of translation.

The prior network and recognition network with an identical PointNet structure consisting of three FC layers of output dimensions (64, 128, 512), followed by another FC layer to generate an 8-dim latent variable. To avoid posterior collapse, a PointNet structure with channel dimensions (8, 8, 8) in the context encoder.

The prediction network concatenates the generated latent variable and context features and feeds them into subsequent FC layers of channels (64, 64) before predicting offsets and directions.

4.4 Traing and Inference Strategies

To optimize GLENet, the Adam optimizer is used with a learning rate of 0.003, β_1 of 0.9, and β_2 of 0.99. The model was trained for a total of 400 epochs on the KITTI dataset and 40 epochs on the Waymo dataset, with a batch size of 64 on two GPUs. The one-cycle policy was used to update the learning rate. During training, common data augmentation techniques, such as random flipping, scaling (scaling factor uniformly drawn from $[0.95, 1.05]$), and rotation (rotation angle uniformly drawn from $[-\pi/4, \pi/4]$), were applied. To account for incomplete point clouds, an occlusion-driven augmentation approach is proposed, where a complete point cloud may resemble an incomplete one, but their ground-truth bounding boxes differ significantly. To mitigate posterior collapse, KL annealing is used to gradually increase the weight of the KL loss from 0 to 1. k-fold cross-sampling is used dividing the training objects into ten mutually exclusive subsets. During each iteration, GLENet was trained on 9 subsets and made predictions on the remaining subset to generate label uncertainty estimates for the entire training set. During inference, the latent variable z is sampled from the predicted prior distribution $p_\theta(z|c)$ 30 times to form multiple predictions, and the variance of these predictions was used as the label uncertainty.

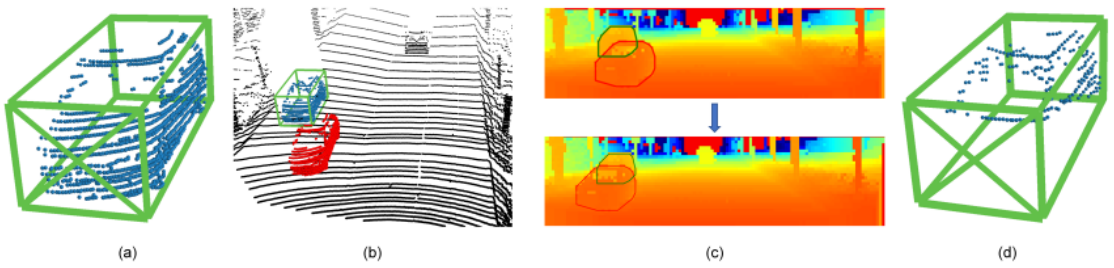


FIGURE 4.1: (a) Original point cloud with annotated ground-truth bounding box (b) the object in red in front of original object in blue (c) Based on the red object, the occluded region of the blue object is estimated (d) Final augmented object with the annotated ground-truth bounding boxes

4.5 Base Detectors

Integrated the GLENet into four popular deep 3D object detection frameworks, i.e., SECOND, CIA-SSD, CenterPoint (two-stage) and Voxel R-CNN, to construct probabilistic detectors, which are dubbed as GLENet-S, GLENet-C, GLENet-CP, and GLENet-VR, respectively. Specifically, we introduced an extra FC layer on the top of the detection head to estimate standard deviations along with the box locations. Meanwhile, applied the proposed UAQE to GLENet-VR to facilitate the training of the IoU-branch. We have set the value of σ_t to 0.05 and the value of μ to 0.01 in KITTI and 0.7 in Waymo dataset in 3D variance voting.

4.6 KITTI Dataset

Method	Reference	Modality	3D AP_{R40}			
			Easy	Mod.	Hard	mAP
MV3D (Chen et al., 2017)	CVPR'17	RGB+LiDAR	74.97	63.63	54.00	64.20
F-PointNet (Qi et al., 2018)	CVPR'18	RGB+LiDAR	82.19	69.79	60.59	70.86
MMF (Liang et al., 2019)	CVPR'19	RGB+LiDAR	88.40	77.43	70.22	78.68
PointPainting (Vora et al., 2020)	CVPR'20	RGB+LiDAR	82.11	71.70	67.08	73.63
CLOCs (Pang et al., 2020)	IROS'20	RGB+LiDAR	88.94	80.67	77.15	82.25
EPNet (Huang et al., 2020)	ECCV'20	RGB+LiDAR	89.81	79.28	74.59	81.23
3D-CVF (Yoo et al., 2020)	ECCV'20	RGB+LiDAR	89.20	80.05	73.11	80.79
STD (Yang et al., 2019)	ICCV'19	LiDAR	87.95	79.71	75.09	80.92
Part-A2 (Shi et al., 2020b)	TPAMI'20	LiDAR	87.81	78.49	73.51	79.94
3DSSD (Yang et al., 2020)	CVPR'20	LiDAR	88.36	79.57	74.55	80.83
SA-SSD (He et al., 2020)	CVPR'20	LiDAR	88.80	79.52	72.30	80.21
PV-RCNN (Shi et al., 2020a)	CVPR'20	LiDAR	90.25	81.43	76.82	82.83
PointGNN (Shi and Rajkumar, 2020b)	CVPR' 20	LiDAR	88.33	79.47	72.29	80.03
Voxel-RCNN (Deng et al., 2021)	AAAI'21	LiDAR	90.90	81.62	77.06	83.19
SE-SSD (Zheng et al., 2021b)	CVPR'21	LiDAR	91.49	82.54	77.15	83.73
VoTR (Mao et al., 2021b)	ICCV'21	LiDAR	89.90	82.09	**79.14**	83.71
Pyramid-PV (Mao et al., 2021a)	ICCV'21	LiDAR	88.39	82.08	77.49	82.65
CT3D (Sheng et al., 2021)	ICCV'21	LiDAR	87.83	81.77	77.16	82.25
GLENet-VR (Ours)	-	LiDAR	**91.67**	**83.23**	78.43	**84.44**

FIGURE 4.2: Comparison of SoTA methods on KITTI test set for vehicle detection, under the evaluation metric of 3D Average Precision (AP) of 40 sampling recall points.

4.7 Waymo Open Dataset

This superior performance can be attributed to the effective handling of bounding box ambiguity, including distant and sparse point cloud objects.

Methods	Reference	3D AP_{R11} Easy	3D AP_{R11} Moderate	3D AP_{R11} Hard	3D AP_{R40} Easy	3D AP_{R40} Moderate	3D AP_{R40} Hard
Part-A^2 (Shi et al., 2020b)	TPAMI'20	89.47	79.47	78.54	-	-	-
3DSSD (Yang et al., 2020)	CVPR'20	89.71	79.45	78.67	-	-	-
SA-SSD (He et al., 2020)	CVPR'20	90.15	79.91	78.78	92.23	84.30	81.36
PV-RCNN (Shi et al., 2020a)	CVPR'20	89.35	83.69	78.70	92.57	84.83	83.31
SE-SSD (Zheng et al., 2021b)	CVPR'21	**90.21**	85.71	**79.22**	93.19	**86.12**	83.31
VoTR (Mao et al., 2021b)	ICCV'21	89.04	84.04	78.68	-	-	-
Pyramid-PV (Mao et al., 2021a)	ICCV'21	89.37	84.38	78.84	-	-	-
CT3D (Sheng et al., 2021)	ICCV'21	89.54	86.06	78.99	92.85	85.82	83.46
SECOND (Yan et al., 2018b)	Sensors'18	88.61	78.62	77.22	91.16	81.99	78.82
GLENet-S (Ours)	-	88.68	82.95	78.19	91.73	84.11	81.35
CIA-SSD (Zheng et al., 2021a)	AAAI'21	90.04	79.81	78.80	**93.59**	84.16	81.20
GLENet-C (Ours)	-	89.82	84.59	78.78	93.20	85.16	81.94
Voxel R-CNN (Deng et al., 2021)	AAAI'21	89.41	84.52	78.93	92.38	85.29	82.86
GLENet-VR (Ours)	-	89.93	**86.46**	79.19	93.51	86.10	**83.60**

FIGURE 4.3: Performance Comparison of 3D Average Precision with 11 sampling recall points on KITTI dataset.

Method	Pedestrian Easy	Pedestrian Moderate	Pedestrian Hard	Cyclist Easy	Cyclist Moderate	Cyclist Hard
Second	56.55	**52.97**	47.73	80.59	67.14	63.11
GLENet-S	**58.22**	52.39	**49.53**	**82.67**	**68.29**	**65.62**
Voxel R-CNN	**66.32**	60.52	55.42	86.62	70.69	66.05
GLENet-VR	66.18	**62.05**	**56.00**	**87.28**	**74.07**	**70.90**

FIGURE 4.4: Performance Comparison on the KITTI val set for pedestrian and cyclist class using AP_{R11}

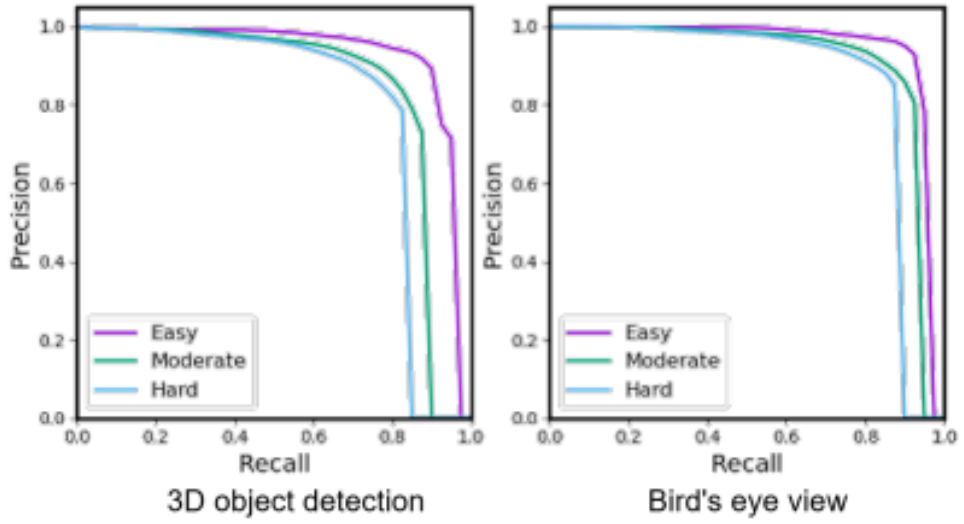


FIGURE 4.5: PR curves of GLENet-VR on the car class of the KITTI test set

Methods	LEVEL_1 3D mAP				mAPH	LEVEL_2 3D mAP				mAPH
	Overall	0-30m	30-50m	50m-inf	Overall	Overall	0-30m	30-50m	50m-inf	Overall
PointPillar (Lang et al., 2019)	56.62	81.01	51.75	27.94	-	-	-	-	-	-
MVF (Zhou et al., 2020)	62.93	86.30	60.02	36.02	-	-	-	-	-	-
PV-RCNN (Shi et al., 2020a)	70.30	91.92	69.21	42.17	69.69	65.36	91.58	65.13	36.46	64.79
VoTr-TSD (Mao et al., 2021b)	74.95	92.28	73.36	51.09	74.25	65.91	-	-	-	65.29
Pyramid-PV (Mao et al., 2021a)	76.30	92.67	74.91	54.54	75.68	67.23	-	-	-	66.68
CT3D (Sheng et al., 2021)	76.30	92.51	75.07	55.36	-	69.04	91.76	68.93	42.60	-
SECOND* (Yan et al., 2018b)	69.85	90.71	68.93	41.17	69.40	62.76	86.92	62.57	35.89	62.30
GLENet-S (Ours)	72.29	91.02	71.86	45.43	71.85	64.78	87.56	65.11	38.60	64.25
CenterPoint-TS* (Yin et al., 2021)	75.52	92.09	74.35	54.27	75.07	67.37	90.89	68.11	42.46	66.94
GLENet-CP (Ours)	76.73	92.70	75.70	55.77	76.27	68.50	91.95	69.43	43.68	68.08
Voxel R-CNN* (Deng et al., 2021)	76.08	92.44	74.67	54.69	75.67	68.06	91.56	69.62	42.80	67.64
GLENet-VR (Ours)	**77.32**	**92.97**	**76.28**	**55.98**	**76.85**	**69.68**	**92.09**	**71.21**	**44.36**	**68.97**

FIGURE 4.6: Comparison on Waymo Validation set for vehicle detection

Chapter 5

Ablation Study

5.1 Comparison with Other Label Uncertainty Estimation

I compared GLENet with two other ways of label uncertainty estimation: 1) treating the label distribution as the deterministic Dirac delta distribution with zero uncertainty 2) estimating the label uncertainty with simple heuristics, i.e., the number of points in the ground-truth bounding box or the IoU between the label bounding box and its convex hull of the aggregated LiDAR observations

Methods	3D AP_{R40} Easy	Moderate	Hard
Voxel R-CNN	92.38	85.29	82.86
GLENet-VR w/ L_{KLD} (σ^2=0)	92.48	85.37	83.05
GLENet-VR w/ L_{KLD} (points num)	92.46	85.58	83.16
GLENet-VR w/ L_{KLD} (convex hull)	92.33	85.45	82.81
GLENet-VR w/ L_{KLD} (Ours)	**93.49**	**86.10**	**83.56**

FIGURE 5.1: Comparison of different label uncertainty estimation approaches

Method	AP_{BEV} for IoU@0.7		
	Easy	Mod.	Hard
PIXOR (Yang et al., 2018)	86.79	80.75	76.60
ProbPIXOR + $\mathcal{L}_{KLD}$ ($\sigma = 0$)	88.60	80.44	78.74
ProbPIXOR + $\mathcal{L}_{KLD}$ (Wang et al., 2020)	**92.22**	82.03	79.16
ProbPIXOR + $\mathcal{L}_{KLD}$ (Ours)	91.50	**84.23**	**81.85**

FIGURE 5.2: Comparison on KITTI dataset

5.2 Key Components of Probabilistic Detectors

Only training with the KL loss brings little performance gain. Introducing the label uncertainty generated by GLENet into the KL Loss contributes improvements which demonstrates its regularization effect on KL-loss and its ability to estimate more reliable uncertainty statistics of bounding box labels. The UAQE module validates its effectiveness in estimating the localization quality.

Observed that the UAQE module effectively reduces the IoU estimation error across various intervals of actual IoU values. It demonstrates that the UAQE module not only improves the overall average precision (AP) metric but also enhances the accuracy of location quality estimation.

KL loss	LU	var voting	UAQE	Easy	Moderate	Hard
				92.38	85.29	82.86
✓				92.45	85.25	82.99
✓		✓		92.48	85.37	83.05
✓	✓			93.20	85.76	83.29
✓	✓	✓		93.24	85.91	83.41
✓	✓	✓	✓	**93.49**	**86.10**	**83.56**

FIGURE 5.3: Contribution of each component in our constructed GLENet-VR pipeline

5.3 Ablation Study of GLENet

Effectiveness of Preprocessing To eliminate the local impact of translation on GLENet's input, the point cloud of a single object was standardized to zero mean. However, this process might remove meaningful distance information. Distant objects with fewer points typically have high label uncertainty, while closer objects

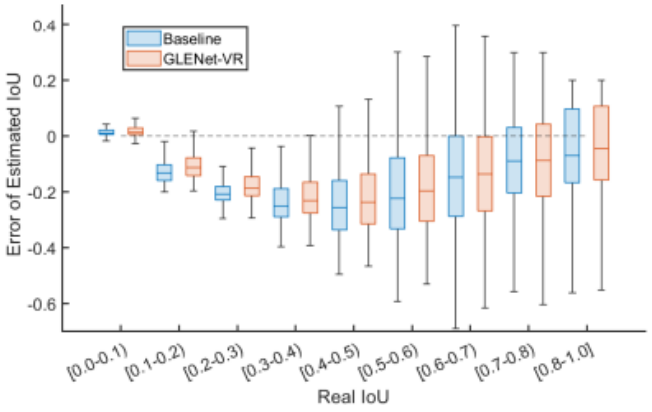


FIGURE 5.4: Boxplots are used to display the estimated IoU error across various intervals of true IoU values. The x-axis represents the real IoU between proposals and their corresponding GT boxes, while the y-axis represents the distribution of estimation error, which is the difference between the estimated IoU score and the real IoU

usually have a high point count and low label uncertainty. To investigate this, we performed experiments by adding the absolute coordinates of the point cloud as an extra feature to GLENet's input.

NC	AC	$L_{NLL}\downarrow$	Easy	Mod.	Hard	Avg
✓		**91.50**	**93.49**	**86.10**	**83.56**	**87.72**
✓	✓	147.33	93.21	85.66	83.35	87.41

FIGURE 5.5: Effect of point cloud input with and without absolute coordinates in GLENet

As shown in Table 8, including extra absolute coordinates did not yield significant improvements in the negative log-likelihood (L_{NLL}) metric or the performance of downstream detectors. We reason these observations from two aspects. First, the additional absolute coordinates may differentiate objects located at different positions but with similar appearances, resulting in fewer samples with similar shapes but different bounding box labels. This makes it difficult for GLENet to capture the one-to-many relationship between incomplete point clouds and plausible bounding boxes. Second, the absolute distance and point cloud density are generally correlated, i.e., an object with a larger absolute distance generally has a sparser point cloud representation, and this correlation could be perceived by the network. In

other words, the absolute distance information is somewhat redundant to the network.

Necessity of Context Encoder In addition to learning the distribution of latent variables, the prior and recognition networks are also capable of extracting features from point clouds. To verify the necessity of the context encoder that is responsible for encoding contextual information from the input data in GLENet, we conducted an ablation experiment.

Setting	$L_{NLL}\downarrow$	Easy	Mod.	Hard	Avg.
Baseline	**91.50**	**93.49**	**86.10**	**83.56**	**87.72**
w/o Occlusion Augmentation	230.10	92.96	85.52	83.07	87.18
w/o Context Encoder	434.93	92.65	85.31	82.59	86.85

FIGURE 5.6: Occlusion augmentation techniques and context encoder in GLENet for evaluation of GLENet and the 3D average precisions of 40 sampling recall points for evaluation of downstream detectors

As shown in Table 9, after removing the context encoder, we observed a significant deterioration in both the L_{NLL} metric and the average precision (AP) of the downstream detector. These results clearly demonstrate the necessity of the context encoder to extract geometric features from point clouds and allow the recognition and prior networks to focus on capturing the underlying structure of the input data in a low-dimensional space. Without the context encoder, the recognition and prior networks would need to learn both the geometric features and the contextual information from the input data, which would lead to poorer performance.

5.4 Dimension of the Latent Variable

Table. 10 shows the performance of adopting latent variables with various dimensions for GLENet. We can observe that the accuracy increase gradually, with the dimensions of latent variables from 2 to 8, and the setting of 32-dimensional latent variables achieve similar performance.

The results demonstrate a too-small dimension of the latent variables makes the GLENet unable to fully represent the underlying structure of the input data. And setting the dimension of latent variables to larger values like 64 or 128 can lead

Dimensions	$L_{NLL}\downarrow$	Easy	Mod.	Hard	Avg.
2	856.48	92.05	84.69	82.22	86.32
4	605.11	92.25	85.11	82.24	86.53
8	91.50	**93.49**	**86.15**	83.56	**87.73**
32	**86.16**	93.28	85.94	**83.60**	87.60
64	110.49	93.11	85.51	83.27	87.30
128	105.93	92.74	85.82	83.10	87.22

FIGURE 5.7: Ablation study of the dimensions of latent variables in GLENet.

to over-fitting and slight decreases in performance. When the dimension of the latent variables is too large, the model can easily memorize the noise and details in the training data, which is not helpful for generating new and useful samples. Besides, though the setting of 32-dim latent variables leads to the lowest L_{NLL}, the performance of downstream detectors is best using label uncertainty with 8-dim latent variables. Therefore, though the L_{NLL} metric can reflect the quality of generating of GLENet to some extent, it is not guaranteed to be strongly correlated with the performance of downstream detectors.

Effects of the Sampling Times In Table 11, we investigate the effects of the sampling times to calculate label uncertainty. We can observe that larger sampling times generally achieve lower L_{NLL} and better performance of downstream detectors, and similar performance is observed when using more than 30 sampling times.

Times	$L_{NLL}\downarrow$	Easy	Mod.	Hard	Avg.
4	608.82	92.54	85.11	81.21	86.29
8	240.08	92.96	85.52	82.80	87.09
16	148.21	92.99	85.66	83.35	87.33
30	91.5	93.49	86.10	**83.56**	**87.72**
64	86.76	93.37	**86.16**	83.42	87.65
128	**77.06**	**93.53**	85.92	83.47	87.64

FIGURE 5.8: Ablation study of the sampling times to calculate label uncertainty in GLENet.

Statistically speaking, the variance obtained after a certain number of sampling times will tend to stabilize. Hence, to balance the computation cost and performance, we empirically choose to calculate the label uncertainty with predicted multiple bounding boxes by sampling the latent variables 30 times.

Methods		Voxel R-CNN (Deng et al., 2021)	GLENet-VR (Ours)	Improvement
Occlusion[b]	0	92.35	93.51	+1.16
	1	76.91	78.64	+1.73
	2	54.32	56.93	+2.61
Distance	0-20m	96.42	96.69	+0.27
	20-40m	83.82	86.87	+3.05
	40m-Inf	38.86	39.82	+0.96

FIGURE 5.9: Comparison on different occlusion levels and distance ranges, evaluated by the 3D Average Precision (AP) calculated with 40 sampling recall positions on the KITTI val set

Method	FPS (Hz)
SECOND Yan et al. (2018b)	23.36
GLENet-S (Ours)	22.80
CIA-SSD Zheng et al. (2021a)	27.18
GLENet-C (Ours)	28.76
Voxel R-CNN Deng et al. (2021)	21.08
GLENet-VR (Ours)	20.82

FIGURE 5.10: Inference time comparison for different baselines on the KITTI dataset.

5.5 Comparison of Visual Results

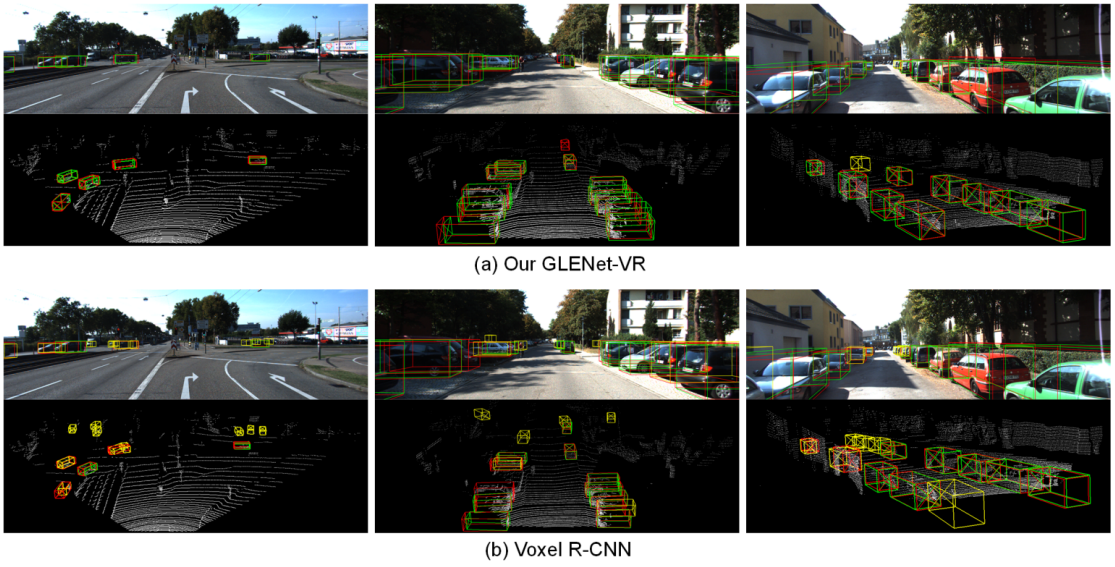
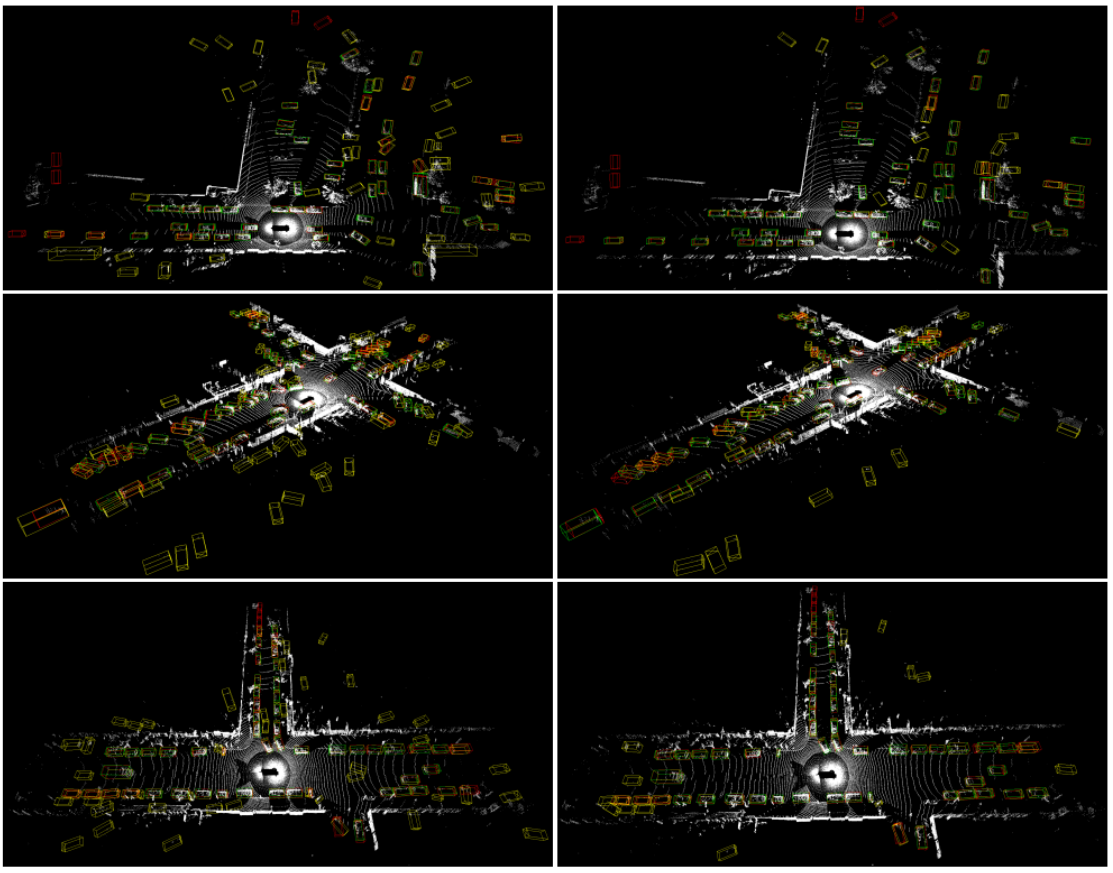


FIGURE 5.11: Visual comparison of the results by GLENet-VR and Voxel R CNN on the KITTI dataset. The ground-truth, true positive and false positive bounding boxes are visualized in red, green and yellow, respectively



(a) SECOND (b) GLENet-S (Ours)

FIGURE 5.12: Visual comparison of the results by SECOND and GLENet-S on the Waymo val set. Additional NMS with a higher IoU threshold is conducted to eliminate overlapped bounding boxes for better visualization.

Chapter 6

Discussion

6.1 Limitations

(1) **Complexity and Computational Cost:** While GLENet provides reliable label uncertainty as supervision signals for downstream probabilistic detectors, estimating the label uncertainty itself increases computational costs and overall training complexity. Particularly, to mitigate overfitting, k-fold cross-sampling was employed, training GLENet on 9 subsets and making predictions on the remaining subset at each iteration.

(2) **Incomplete Input Information:** In GLENet, only the partial point cloud of individual objects is used as input, utilizing solely the learned geometric information to estimate potential bounding boxes. However, context cues like free space and the location of surrounding objects, which are also meaningful for determining bounding boxes, are neglected. While incorporating such information could be beneficial, it may compromise GLENet's core benefit of learning the latent distribution of bounding boxes from samples with similar point cloud shapes, as involving the whole point cloud scene would distinguish objects with similar shapes.

(3) **Robustness to Annotation Errors:** Although GLENet aims to address inherent ambiguity in ground-truth annotations, it may not be entirely immune to significant annotation errors. If the training data contains substantial errors, the model may inadvertently learn and propagate these errors, leading to inaccurate

label uncertainty estimation. For instance, if an object with a high-quality point cloud is annotated with a wrong box, resulting in inconsistent predictions and larger label uncertainty, objects with similar shapes will suffer from unreasonable label uncertainty supervision signals.

(4) **Limited Evaluation Metrics and Scenarios:** Evaluating the quality and diversity of generated data in generative tasks like GLENet is challenging. While the proposed L_{NLL} assesses the closeness between GLENet's predictions and ground-truth annotation bounding boxes, evaluating the quality and diversity of generated data remains an ongoing research problem. Additionally, while performance gains are demonstrated on benchmark datasets like KITTI and Waymo, the generalizability of the approach across various environmental conditions, object classes, and sensor modalities could be a limitation.

(5) **Possible Extensions:** The idea of estimating label uncertainty by capturing the one-to-many relationship between observed input and multiple plausible labels with latent variables could be extended to other subjective tasks in computer vision where labels are not deterministic. Promising tasks include 3D object tracking, where different annotator opinions on object boundaries lead to non-deterministic labels, and image quality assessment, where the goal is to evaluate the subjective quality of an image, often in the context of compression or transmission.

6.2 Summary

Introduced a general and unified deep learning-based approach for modeling 3D object-level label uncertainty. Specifically, we proposed GLENet, adapted from the Conditional Variational Autoencoder (CVAE) learning framework, to capture the one-to-many relationships between incomplete point cloud objects and potentially plausible bounding boxes. As a plug-and-play component, GLENet can generate reliable label uncertainty statistics that can be seamlessly integrated into various 3D detection pipelines to build powerful probabilistic detectors. We validated the effectiveness and versatility of our method by incorporating the proposed GLENet into several existing deep 3D object detectors, which consistently improved their

performance and achieved state-of-the-art results on both the KITTI and Waymo datasets.

6.3 Future Work

A lot of work is to be done in the incident detection problem in traffic video:

- Integrating intention of target vehicles and pedestrians in the network through motion prediction for insurance disputes
- Add information from sound
- Add human action classifier to predict any signs given by the pedestrians

Bibliography

[1] Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S.: Learning temporal regularity in video sequences. In: Procs. of the IEEE Conf. on computer vision and pattern recognition. available: https://openaccess.thecvf.com/content_cvpr_2016/papers/Hasan_Learning_Temporal_Regularity_CVPR_2016_paper.pdf.

[2] Luo, W., Liu, W., Gao, S.: Remembering history with convolutional lstm for anomaly detection. In: 2017 IEEE Intl. Conf. on Multimedia and Expo. available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8019325.

[3] Wang, L., Zhou, F., Li, Z., Zuo, W., Tan, H.: Abnormal event detection in videos using hybrid spatio-temporal autoencoder. available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8451070.

[4] Liu, W., Luo, W., Lian, D., Gao, S.: Future frame prediction for anomaly detection–a new baseline. In: Proc. of Conf. on Computer Vision and Pattern Recognition available: https://arxiv.org/pdf/1712.09867.pdf.

[5] Zhou, Z., Dong, X., Li, Z., Yu, K., Ding, C., Yang, Y.: Spatio-Temporal Feature Encoding for Traffic Accident Detection in VANET Environment. IEEE Tran. on Intelligent Transportation Systems. available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9714213.

[6] Yao, Y., Wang, X., Xu, M., Pu, Z., Wang, Y., Atkins, E., Crandall, D.: Dota: Unsupervised detection of traffic anomaly in driving videos. IEEE Tran. on Pattern Analysis and Machine Intelligence. available: http://vision.soic.indiana.edu/papers/dota2022pami.pdf.

[7] Xu, M., Gao, M., Chen, Y.T., Davis, L.S., Crandall, D.J.: Temporal recurrent networks for online action detection. In: Procs. of the IEEE/CVF Intl. Conf. on Computer Vision. available: `https://arxiv.org/pdf/1811.07391.pdf`.

[8] Leonardo R., Vittorio B., Tomasao F., Massimo B., Andrea P.: Memory-Augmented online video anomaly detection. available: `https://arxiv.org/pdf/2302.10719v2.pdf`.

[9] Dataset. available : `https://waymo.com/open/data/perception` and `http://www.cvlibs.net/datasets/kitti`