# RCDN: Towards Robust Camera-Insensitivity Collaborative Perception via Dynamic Feature-based 3D Neural Modeling

**Tianhang Wang**
Tongji University
tianya_wang@tongji.edu.cn

**Fan Lu**
Tongji University
lufan@tongji.edu.cn

**Zehan Zheng**
Tongji University
zhengzehan@tongji.edu.cn

**Guang Chen***
Tongji University
guangchen@tongji.edu.cn

**Changjun Jiang**
Tongji University
cjjiang@tongji.edu.cn

## Abstract

Collaborative perception is dedicated to tackling the constraints of single-agent perception, such as occlusions, based on the multiple agents' multi-view sensor inputs. However, most existing works assume an ideal condition that all agents' multi-view cameras are continuously available. In reality, cameras may be highly noisy, obscured or even failed during the collaboration. In this work, we introduce a new robust camera-insensitivity problem: *how to overcome the issues caused by the failed camera perspectives, while stabilizing high collaborative performance with low calibration cost?* To address above problems, we propose **RCDN**, a **R**obust **C**amera-insensitivity collaborative perception with a novel **D**ynamic feature-based 3D **N**eural modeling mechanism. The key intuition of RCDN is to construct collaborative neural rendering field representations to recover failed perceptual messages sent by multiple agents. To better model collaborative neural rendering field, RCDN first establishes a geometry BEV feature based time-invariant static field with other agents via fast hash grid modeling. Based on the static background field, the proposed time-varying dynamic field can model corresponding motion vectors for foregrounds with appropriate positions. To validate RCDN, we create OPV2V-N, a new large-scale dataset with manual labelling under different camera failed scenarios. Extensive experiments conducted on OPV2V-N show that RCDN can be ported to other baselines and improve their robustness in extreme camera-insensitivity settings.

## 1 Introduction

Multi-agent collaborative perception[1–5] obtains better and more holistic perception by allowing multiple agents to exchange complementary perceptual information. This field has the potential to effectively address various persistent challenges in single-perception, such as occlusion[6, 7]. The associated techniques and systems also process significant promise in various domains, such as the utilization of multiple unmanned aerial aircraft for search and rescue operations[8–10], the automation and mapping of multiple robots[11–13]. As an emerging field, the research of collaborative perception faces several issues that need to be addressed. These challenges include the need for high-quality datasets[14–17], the formulation of models that are agnostic to specific tasks and models[18, 19], and the ability to handle pose error and adversarial attacks[20, 21].

*Corresponding Author. Our code is available at: https://github.com/ispc-lab/RCDN.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

However, a vast majority of existing works do not seriously account for the harsh realities[22, 23] of real-world sensors in the collaboration, such as blurred, high noise, interruption and even failure. These factors directly undermine the basic collaboration premise[24, 25] of reconstructing the holistic view based on the multi-view sensors that severely impact the reliability and quality of collaborative perception process. This raises a critical inquiry: *how to overcome the issues caused by the failed cameras' perspectives while stabilizing high collaborative performance with low calibration cost?* The designation *camera insensitivity* overcomes the unpredictable essence of the specific failure camera numbers and time; see Figure 1 for an illustration. To address this issue, one viable solution is adversarial defense[26]. By robust defense strategy, adversarial defense bypasses camera insensitivity among blurred and noise. However, its performance is suboptimal[27] and has been shown to be particularly vulnerable to noise ratios[20] and failed camera numbers.

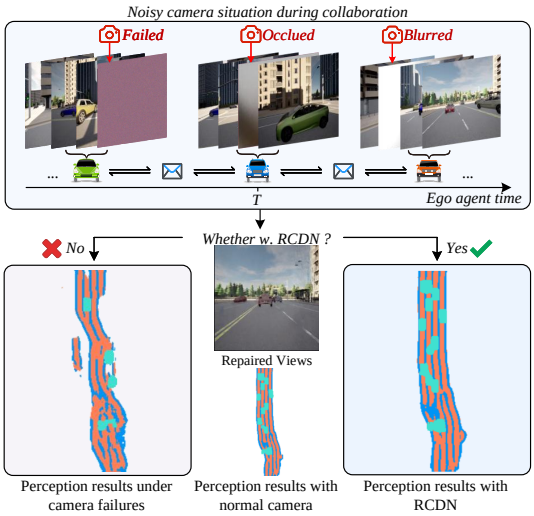



Figure 1: Illustration of noisy camera situations (blurred, occluded and even failed) during collaboration and the perception result *w.o./w.* RCDN. orange for drivable areas segmentation, blue for lanes and teal for dynamic vehicles.

To address this robust camera insensitivity collaborative perception problem, we propose **RCDN**, a **R**obust **C**amera-insensitivity collaborative perception with a **D**ynamic feature-based 3D **N**eural modeling mechanism. The core idea is to recover noisy camera perceptual information from other agents' views by modeling the collaborative neural rendering field representations. Specifically, RCDN has two collaborative field phases: a time-invariant static background field and time-varying dynamic foreground field. In the static phases, RCDN sets other baselines' backbone as the collaboration base and undertakes end-to-end training to create a robust unified geometry Bird-eye view (BEV[28, 29]) feature space for all agents. Then, the geometry BEV feature combines the hash grid modeling, an explicit and multi-resolution network, to generate static background views through $\alpha$-composed accumulation of RGB values along a ray at a fast speed. In the dynamic phase, RCDN utilizes 4D spatiotemporal position features to model the dynamic motion of 3D points, which learns an accurate motion field under optical priors and spatiotemporal regularization. The proposed RCDN has two major advantages: i) RCDN can handle camera insensitivity collaboration under unknown noisy timestamps and numbers; ii) RCDN does not put any extra communication burden into inference stage and costs little computation burden.

In our efforts to validate the effectiveness of RCDN, we identified a gap: the lack of a comprehensive collaborative perception dataset that accounts for different camera noise scenarios. To address this, we create the OPV2V-N, an expansive new dataset derived from OPV2V, featuring meticulously labeled timestamps and camera IDs. This advancement aims to support and enhance research in camera-insensitive collaborative perception. Extensive experiments on OPV2V-N show RCDN's remarkable performance when other baselines equipped with RCDN under extreme camera-insensitivity setting, improving *w.o.* RCDN baseline methods by about $157.91\%$.

## 2 Related Works

**Robust Single Perception.** Single-agent perceptions[30, 31, 27, 32–34] have tackled the robust camera setting with other sensor modals. [27] reveals that camera-based methods [34] can be easily effected by camera working conditions. Some works[32, 31] introduce LiDAR into perception system and design a soft-association mechanism between the LiDAR and the inferior camera-side, to relieve the negative impacts caused by cameras. MVX-Net[33] improves the combination pipeline of LiDAR and cameras by leveraging the VoxelNet[35] architecture. CRN[30] introduces the low-cost Radar to replace the LiDAR, which can provide precise long-range measurement and operates reliably in all environments. However, as for the camera-only situation, few work seeks to solve this because recovering just from the single-view is highly ill-posed (with infinitely many solutions that match the

input image). With the recent rapid development of V2X[36], we now can introduce the multi-agent and multi-view based collaborative perception setting to explore this extreme situation.

**Collaborative Perception.** Perception tasks for single agents can be adversely affected by factors such as limited sensor fields of view and physical ambient occlusions. To address the aforementioned challenges, collaborative perception[37–39] can attain more comprehensive perceptual output by exchanging perception data. Early techniques involved the transmission of either unprocessed sensory input (referred to as early fusion) or the results of perception (referred to as late fusion). Nevertheless, recent research has been examining the transfer of intermediate features to achieve a balance between performance and bandwidth. Some works[40–43] devote selecting the most informative messages to communicate. DiscoNet[44] utilizes knowledge distillation to achieve a better trade-off between performance and bandwidth. V2X-ViT[45] presents a unified V2X framework based on Transformer that takes into account the heterogeneity of V2X system. Meanwhile, some learnable or mathematical based methods[46–49] have also been proposed to correct the pose errors and latency. Moreover, some works[50, 51] reveal that the holistic character of collaborative perception can improve the effect of driving planning and control tasks. However, most existing papers do not take the harsh realities of real-world sensors into account, such as blurred, high noise, occlusion and even failure, which directly undermine the basic collaboration premise of multi-view based modeling, negatively impacting performance. This work formulates camera-insensitivity collaborative perception, which considers real-world camera sensor conditions.

**Neural Rendering.** Neural radiance fields[52] aim to utilize implicit neural representations to encode densities and colors of the scene. This approach takes advantage of volumetric rendering to synthesize views, and it can be effectively optimized from 2D multi-view images. Hence, numerous works have enhanced NeRF in terms of rendering quality[53–55], efficiency[56–59], *etc*. For example, Mip-NeRF[60] utilizes cone tracing instead of ray tracing in standard NeRF volume rendering by introducing integrated positional encoding, which greatly improves the render quality. To improve the efficiency of training and inference processes, Instant-NGP[61] proposes a learned parametric multi-resolution hash for efficient encoding, which also leads to high compactness. Some works have also extended NeRF to large-scale urban autonomous scenes[62–64]. In this work, we first introduce neural rendering to collaborative perception. The proposed collaborative neural rendering field representations will address the problem of recovering highly noisy perceptual messages.

## 3 Problem Formulation

Consider $N$ agents in a scene, where each agent can send and receive collaboration messages from other agents. For the $n$-th agent, let $\mathcal{X}_n^{t_i} = \{\mathcal{I}_c^{t_i}\}_{c=1}^{c_n}$ and $\mathcal{Y}_n^{t_i}$ be the raw observation and the perception ground-truth at time current $t_i$, respectively, where $\mathcal{I}_c^{t_i}$ is the $c$-th camera images recorded at $i$-th timestamp, and $\mathcal{P}_{m\to n}^{t_i}$ is the collaboration message sent from the agent $m$ at time $t_i$. The key of the camera insensitivity is that the specific noisy camera number and corresponding timestamp are unpredictable. Therefore, each agent has to encounter invalid view information, which contains both local observation and collaboration messages sent from other agents. Then, the task of camera insensitivity collaborative perception is formulated as:

$$\max_{\theta_1,\theta_2,\mathcal{P}} \sum_{n=1}^{N} g\left(\widehat{\mathbf{Y}}_n^{t_i}, \mathbf{Y}_n^{t_i}\right) \tag{1}$$
$$\text{subject to } \widehat{\mathbf{Y}}_n^{t_i} = \boldsymbol{c}_{\theta_2}(\boldsymbol{\pi}_{\theta_1}(\psi(\mathcal{X}_n^{t_i}, \{\mathcal{P}_{m\to n}^{t_i}\}_{m=1}^{N-1}))),$$

where $g(\cdot,\cdot)$ is the perception evaluation metrics, $\widehat{\mathbf{Y}}_n^{t_i}$ is the perception result of the $n$-th agent at time $t_i$, $\psi(\cdot,\cdot)$ is the camera noise function to simulate the harsh realities of the real-world situation, $\boldsymbol{\pi}_{\theta_1}(\cdot)$ is the proposed collaborative neural rendering field network RCDN with trainable parameters $\theta_1$, and $\boldsymbol{c}_{\theta_2}$ is the existing collaborative perception network with trainable parameters $\theta_2$. Note that the proposed RCDN is to recover the noisy camera views caused by the $\psi$ function, making collaborative perception system more robust to the unpredictable situation of noisy camera data.

Given such high noisy camera view, the performances of collaborative perception system would be significantly degraded since the mainstream collaborative perception utilizes the multi-view camera-based BEV features for communication and downstream tasks, and using such damaged features

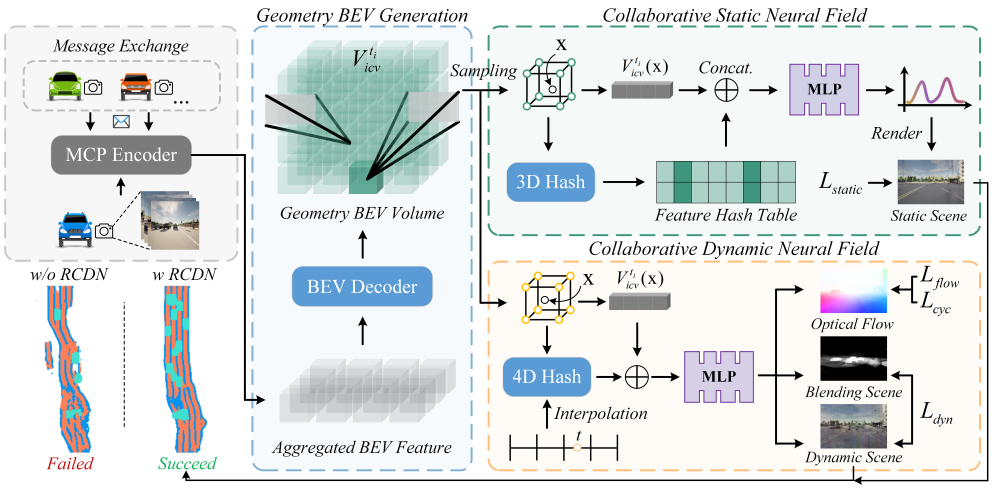



Figure 2: System overview. The geometry BEV generation module provides feature sampling for later processes. The collaborative static and dynamic fields are performed in parallel to model the background and foreground, respectively. Note that MCP is short for the multi-agents collaborative perception process.

would contain erroneous information during the perception process. In the next section, we will introduce RCDN to address this issue.

# 4 RCDN

This section proposes a robust camera-insensitivity collaborative perception system, RCDN. Figure 2 overviews the framework of the RCDN module in Sec.4.1. The details of three key modules of RCDN can be found in Sec.4.2-4.4.

## 4.1 Overall Architecture

The problem of noisy camera view results in the sub-optimization of the holistic multi-view based BEV features generation in the collaboration messages. That is, the collaboration messages from both self and other agents would be noisy or damaged for the fusion process. The proposed RCDN addresses this issue with two key notions: i) we construct novel collaborative neural rendering field representations, enabling collaborative perception to recover from the noisy camera view; and ii) we establish time-invariant and time-varying fields for background and foreground, respectively, making the collaborative neural rendering field more accurate.

Mathematically, let the $n$-th agent be the ego agent and $\mathcal{X}_n^{t_i}$ be its raw observation at the $t_i$ timestamp of agent $n$. The proposed camera-insensitivity collaborative perception system RCDN is formulated as follows:

$$\mathbf{F}_n^{t_i} = f_{\text{enc}}(\psi(\mathcal{X}_n^{t_i}, \{\mathcal{X}_j^{t_i}\}_{j=1}^{N-1})), \tag{2a}$$

$$\mathbf{V}_{icv}^{t_i} = f_{\text{geo_bev}}(\mathbf{F}_n^{t_i}), \tag{2b}$$

$$(\sigma^s, \mathbf{c}^s) = f_{\text{static}}(\mathbf{r}(u_k), \mathbf{V}_{icv}^{t_i}(\mathbf{r}(u_k))), \tag{2c}$$

$$(\mathbf{s}_{fw}, \mathbf{s}_{bw}, \sigma_{t_i}^d, \mathbf{c}_{t_i}^d, \mathbf{b}) = f_{\text{dynamic}}(\mathbf{r}(u_k), \mathbf{V}_{icv}^{t_i}(\mathbf{r}(u_k)), t_i), \tag{2d}$$

$$\widetilde{\mathcal{X}}_n^{t_i}, \{\widetilde{\mathcal{X}}_j^{t_i}\}_{j=1}^{N-1} = f_{\text{render}}(\sigma^s, \mathbf{c}^s, \sigma_{t_i}^d, \mathbf{c}_{t_i}^d, b), \tag{2e}$$

$$\widehat{\mathbf{Y}}_n^{t_i} = f_{\text{mcp}}(\widetilde{\mathcal{X}}_n^{t_i}, \{\widetilde{\mathcal{X}}_j^{t_i}\}_{j=1}^{N-1}), \tag{2f}$$

where $\mathbf{F}_n^{t_i} \in \mathbb{R}^{C\times H\times W}$ is the BEV feature maps of the $n$-th agent at timestamp $t_i$ with $H, W$ the size of BEV map and $C$ the number of channels; $\mathbf{V}_{icv}^{t_i} \in \mathbb{R}^{C\times Z\times H\times W}$ is the implicit collaborative geometry volume feature of the scenarios; which is lifted from BEV plane with the $Z$ height; $\mathbf{r}(u(k))$ is the ray from the failed camera center $\mathbf{o} \in \mathbb{R}^2$ through a given pixel on the image plane as $\mathbf{r}(u(k)) = \mathbf{o} + u(k)\mathbf{d}$, where $\mathbf{d} \in \mathbb{R}^3$ is the normalized viewing direction; $f_{\text{static}}$ is a explicit hash grid based representation to model the collaborative static scenarios volume density $\sigma^s \in \mathbb{R}^1$ and corresponding color $\mathbf{c}^s \in \mathbb{R}^3$; $f_{\text{dynamic}}$ is the dynamic collaborative neural network

takes the interpolated 4D-tuple $(\mathbf{r}(u(k)), t_i)$ and sampled $\mathbf{V}_{icv}^{t_i}$ feature as input and predict 3D collaborative scene flow vectors $\mathbf{s}_{fw}, \mathbf{s}_{bw} \in \mathbb{R}^3$, dynamic volume density $\sigma_{t_i}^d$, color $\mathbf{c}_{t_i}^d$ and blending weight $\mathbf{b} \in \mathbb{R}^2$; and $\widetilde{\mathcal{X}}_n^{t_i}, \{\widetilde{\mathcal{X}}_j^{t_i}\}_{j=1}^{N-1}$ is the recovered noisy camera images at timestamp $t_i$ after collaborative rendering; and $\widehat{\mathbf{Y}}_n^{t_i}$ is the final output of the system. In summary, Step 2a extracts BEV perceptual features from observation data. Step 2b generates the collaborative geometry BEV volume feature map for each timestamp, enabling feature sampling in Step 2c and 2d. Step 2d models the static background field of collaboration scenarios. Step 2d models the dynamic foreground field of collaboration objects. Step 2e gets the global volume density and color information by combining both static and dynamic field models to recover the failed camera perspective images. Finally, Step 2f outputs the final perceptual results with repaired images.

Note that i) Step 2a is done locally, Step 2b-2f are performed after receiving the messages from others. The proposed RCDN does not require any extra transmission during the inference process, which is bandwidth friendly; and ii) Step 2c and 2d are performed in parallel to save inference time; and iii) Same as [44, 49], RCDN adopts the feature representations in bird's eye view (BEV), where the feature maps of all agents are projected to the same global coordinate system. We now elaborate on the details of Steps 2b-2e in the following subsections.

### 4.2 Collaborative Geometry BEV Volume Feature

Given the BEV feature map of each agent, Step 2b aims to construct a unified collaborative geometry BEV volume feature for each timestamp of the scenario. The intuition is that [65] points out that combing with generic feature representations can avoid the per-scene "network memorization" phenomenon[52], which will improve the efficiency of the optimization process. Therefore, using the geometry BEV feature can enable the subsequent Step 2c, 2d to learn more generic networks for both static and dynamic collaborative neural fields, respectively.

To implement, we use a geometry-aware decoder $D_{geo}$ to transform the BEV feature $\mathbf{F}_n^{t_i}$ into the intermediate feature $\mathbf{F}_n^{'t_i} \in \mathbb{R}^{C\times 1\times X\times Y}$ and $\mathbf{F}_{height,n}^{t_i} \in \mathbb{R}^{1\times Z\times X\times Y}$, and this feature is lifted from BEV plane to an implicit collaborative volume feature $\mathbf{V}_{icv}^{t_i} \in \mathbb{R}^{C\times Z\times X\times Y}$:

$$\mathbf{V}_{icv}^{t_i} = \mathrm{sigmoid}(\mathbf{F}_{height,n}^{t_i}) \cdot \mathbf{F}_n^{'t_i}, \tag{3}$$

where $\cdot$ represents dot production along the channel. Eq. 3 lifts the items on the BEV plane into 3D collaborative volume with the estimated height position $\mathrm{sigmoid}(\mathbf{F}_{height,n}^{t_i})$. $\mathrm{sigmoid}(\mathbf{F}_{height,n}^{t_i})$ represents whether there is an item at the corresponding height. Ideally, the collaborative volume feature $\mathbf{V}_{icv}^{t_i}$ contains all the scene items information in the corresponding position.

### 4.3 Static Collaborative Neural Field

After getting the collaborative volume feature $\mathbf{V}_{icv}^{t_i}$, Step 2c aims to construct the background of camera views with the static collaborative neural field. Given an arbitrary 3D scenario position $\mathbf{x} \in \mathbb{R}^3$ and a 2D viewing direction $\mathbf{d} \in \mathbb{R}^2$, we aims to estimate static scenarios volume density $\sigma^s$ and emitted RGB color $\mathbf{c}^s$ using the fast hash grid-based [61] neural network:

$$(\mathbf{c}^s, \sigma^s) = \mathrm{MLP}(\mathbf{G}_\theta^s(\mathrm{contract}(\mathbf{x}), \mathbf{d}); f), \;\; f = \mathbf{V}_{icv}^{t_i}(\mathbf{x}), \tag{4}$$

where $f = \mathbf{V}_{icv}^{t_i}(\mathbf{x})$ is the neural feature trilinearly interpolated from the collaborative geometry BEV volume $\mathbf{V}_{icv}^{t_i}$ at the location $\mathbf{x}$, $\mathbf{G}_\theta^s(\cdot,\cdot)$ is explicit multi-level hash grid representation with the generic $f$ features for fast static collaborative neural field training. Meanwhile, owing to the collaborative scenarios are unbounded, we utilize $\mathrm{contract}(\cdot)$[53] to map 3D scenario position into a bounded ball of radius 2 with regularization, making the estimation optimization process faster and better. Hence, we can compute the color of the pixel (corresponding to the ray $\mathbf{r}(u_k)$ using numerical quadrature for approximating the collaborative volume rendering interval[66]:

$$\mathbf{C}^s(\mathbf{r}) = \sum_{k=1}^{K} T^s(u_k)\alpha^s(\sigma^s(u_k)\delta_k)\mathbf{c}^s(u_k), \tag{5a}$$

$$T^s(u_k) = \exp\left(-\sum_{k'=1}^{k-1} \sigma^s(u_k)\delta_k\right), \tag{5b}$$

where $\alpha^s(x) = 1 - \exp(-x)$ and $\delta_k = u_{k+1} - u_k$ is the distance between two quadrature points. The $K$ quadrature points $\{u_k\}_{k=1}^K$ are drawn uniformly between $u_n$ and $u_f$, which denotes the near and far of the bounded collaborative scenarios. $T^s(u_k)$ indicates the accumulated transmittance from $u_n$ to $u_k$. Here, we denote $\mathbf{r}_i$ as the rays passing through the pixel $i$. Then, the collaborative static neural loss $\mathcal{L}_{static}$ is defined to minimize the $l_2$-loss between the estimated colors $\mathbf{C}^s(\mathbf{r}_i)$ and the ground truth colors $\mathbf{C}^{gt}(\mathbf{r}_i)$ in the static regions (where $\mathbf{M}(\mathbf{r}_i) = 0$):

$$\mathcal{L}_{static} = \sum_i \|\mathbf{C}^s(\mathbf{r}_i) - \mathbf{C}^{gt}(\mathbf{r}_i) \cdot (1 - \mathbf{M}(\mathbf{r}_i))\|_2^2 \tag{6}$$

### 4.4 Dynamic Collaborative Neural Field

While the static collaborative neural field is being modeled, Step 2d is building the dynamic collaborative neural field to construct the foreground of camera views. Our dynamic collaborative neural field takes 4D spatiotemporal position features as input to model dynamic motion of 3D scene flow $\mathbf{s}_{fw}, \mathbf{s}_{bw}$, volume density $\sigma_{t_i}^d$, color $\mathbf{c}_{t_i}^d$ and blending weight $\mathbf{b}$ (Note that blending weights learns how to blend the results from both the static and dynamic collaborative neural fields in an unsupervised manner, avoiding background's structure and appearance conflict the moving objects.):

$$(\mathbf{s}_{fw}, \mathbf{s}_{bw}, \mathbf{c}_{t_i}^d, \sigma_{t_i}^d, \mathbf{b}) = \mathrm{MLP}(\Delta(\mathbf{G}_\theta^d(\mathrm{contract}(\mathbf{x}), \mathbf{d}), t_i); f), \quad f = \mathbf{V}_{icv}^{t_i}(\mathbf{x}), \tag{7}$$

where $G_\theta^d$ shares the same hash grid representations, but for the dynamic collaborative neural field optimization; $\Delta(\cdot, \cdot)$ is the temporal interpolation functions, which makes the $\mathrm{MLP}$ can efficiently learn the features between keyframes in a scalable manner. Meanwhile, to improve the temporal consistency of the proposed field, we compute the collaborative scene flow neighbors $\mathbf{r}(u_k) + \mathbf{s}_{fw}$ and $\mathbf{r}(u_k) - \mathbf{s}_{bw}$ with the predicted collaborative scene flow $\mathbf{s}_{fw}, \mathbf{s}_{bw}$ to warp the collaborative neural field from the neighboring time instance to the current time. Note that the term $\mathbf{s}_{fw}$ stands for forward scene flow, while $\mathbf{s}_{bw}$ refers to backward scene flow. Specifically, the forward scene flow ($\mathbf{s}_{fw}$) estimates the flow from time t to t+1, whereas the backward scene flow ($\mathbf{s}_{bw}$) estimates the flow from time t to t-1. Hence, we can obtain the corresponding density and color of adjacent time by querying the same $\mathrm{MLP}$s model at $\mathbf{r}(u_k) + \mathbf{s}$:

$$(\mathbf{c}_{t_i+1}^d, \sigma_{t_i+1}^d) = \mathrm{MLP}(\Delta(\mathbf{G}_\theta^d(\mathrm{contract}(\mathbf{x} + \mathbf{s}_{fw}), \mathbf{d}), t_i + 1)) \tag{8a}$$

$$(\mathbf{c}_{t_i-1}^d, \sigma_{t_i-1}^d) = \mathrm{MLP}(\Delta(\mathbf{G}_\theta^d(\mathrm{contract}(\mathbf{x} - \mathbf{s}_{bw}), \mathbf{d}), t_i - 1)) \tag{8b}$$

We can compute the color of a dynamic pixel of collaborative view at time $t_i$. Hence, with both the static and dynamic collaborative neural fields model, we can easily compose them into a complete model using the predicted blending weight $\mathbf{b}$ and render full color $\mathbf{C}^{full}(\mathbf{r})$ frames at noisy views and time. We utilize the following approximate of collaborative volume rendering integral:

$$\mathbf{C}_{t_i}^{full}(\mathbf{r}) = \sum_{k=1}^{K} T_{t_i}^{full} \left(\alpha^d(\sigma_{t_i}^d \delta_k)(1 - \mathbf{b})\mathbf{c}_{t_i}^d + \alpha^s(\sigma^s \delta_k)\mathbf{b}\mathbf{c}^s\right) \tag{9}$$

Similar to the static collaborative rendering loss, we train the dynamic collaborative neural model by minimizing the $l_2$ reconstruction loss under time unit $\tau = \{t_i, t_i - 1, t_i + 1\}$:

$$\mathcal{L}_{dyn} = \sum_{t \in \tau} \sum_i \|(\mathbf{C}_t^{full}(\mathbf{r}_i) - \mathbf{C}^{gt}(\mathbf{r}_i))\|_2^2 \tag{10}$$

To reduce the amount of ambiguity caused by the sparse views during collaborative perception process, we construct motion matching loss to constrain the proposed dynamic collaborative neural field. As we do not have direct 3D supervision for predicted collaborative scene flow from the motion $\mathrm{MLP}$ model, we utilize 2D optical flow $\boldsymbol{f}$ as indirect supervision. Specifically, we first use the estimated collaborative scene flow to obtain the corresponding 3D point. Then, we project these 3D points onto the 2D reference frame with $\varphi(\cdot)$ function. Hence, we can compute the projected collaborative scene optical flow and enforce it to match the estimated optical flow as follows:

$$\mathcal{L}_{opt} = \sum_i \left(\varphi(\boldsymbol{s}_{\{bw,fw\}}(\mathbf{r}_i)) - \boldsymbol{f}_{\{bw,fw\}}^{gt}(\mathbf{r}_i)\right) \tag{11}$$

Meanwhile, we also regularize the consistency of the collaborative scene flow by minimizing the cycle consistency loss $\mathcal{L}_{cyc}$. See more details in the Appendix B.7.

Table 1: Map-view segmentation of different baseline methods *w.o/w* the proposed RCDN on the OPV2V-N camera-track with one random noisy camera failure in the testing phase. We report IoU for all classes.

| Model / Metric | Static Part (*Perf. Comparison*) | | | | Dynamic Part Vehicle | |
|---|---|---|---|---|---|---|
| | Drivable Area | | Lane | | | |
| | Normal | Failure *w.o/w.* RCDN | Normal | Failure *w.o/w.* RCDN | Normal | Failure *w.o/w.* RCDN |
| F-Cooper[1] | 45.44 | 28.87/44.89(↑55.49%) | 33.17 | 15.95/32.23(↑102.07%) | 63.33 | 29.70/61.76(↑107.95%) |
| AttFuse[16] | 45.59 | 27.99/44.38(↑58.56%) | 33.76 | 18.77/31.50(↑67.82%) | 54.14 | 24.76/52.15(↑110.62%) |
| DiscoNet[44] | 42.30 | 24.31/38.54(↑58.54%) | 24.24 | 12.29/22.97(↑86.90%) | 46.56 | 9.25/43.03(↑365.19%) |
| V2VNet[37] | 41.70 | 27.99/39.72(↑41.91%) | 27.14 | 10.52/25.24(↑139.92%) | 42.57 | 11.28/42.76(↑279.08%) |
| CoBEVT[6] | 51.96 | 32.08/47.19(↑47.10%) | 34.19 | 14.45/29.55(↑104.50%) | 56.61 | 32.41/55.10(↑70.01%) |

### 4.5 Training Details and Optimization

To train the overall system, we supervise two tasks: static and dynamic collaborative neural fields, respectively. Meanwhile, during the training process, the static collaborative field and dynamic collaborative field are trained separately. The initial learning rate is 5e-4 with the exponential learning rate decay strategy. The weight values are set to 1.0, 1.0, 0.1, and 1.0, respectively:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{static} + \lambda_2 \mathcal{L}_{dyn} + \lambda_3 \mathcal{L}_{opt} + \lambda_4 \mathcal{L}_{cyc} \tag{12}$$

## 5 Experimental Results

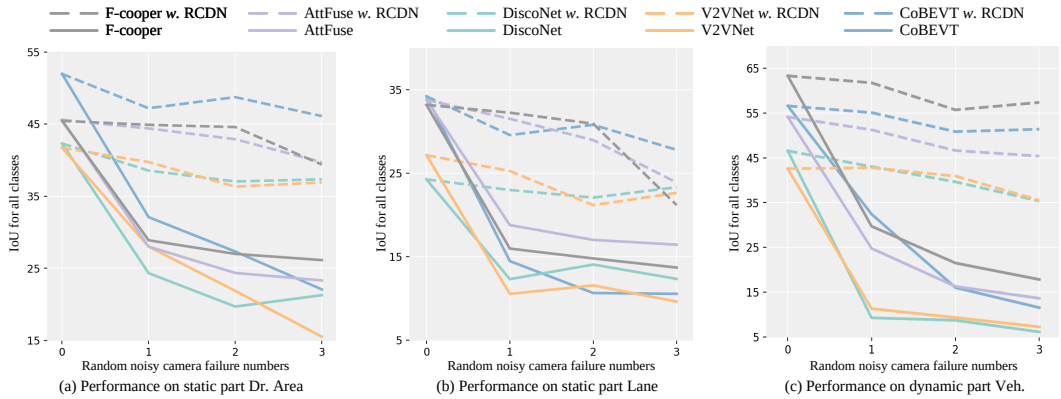
We create the first camera-insensitivity collaborative perception dataset and conduct extensive experiments on OPV2V-N. To ensure the consistency of the input noisy camera data and verify the effectiveness of RCDN, we set the noisy camera data to be in the failed situation[27]. Meanwhile, the task of the experiments is map segmentation, including the performance of the vehicle, drivable area (Dr. area) and lane, totaling three classes. We utilize the Intersection over Union (IoU) between map prediction and ground truth map-view labels as the performance metric.

### 5.1 Datasets

**OPV2V-N**. To facilitate research on camera-insensitivity for collaborative perception, we propose a simulation dataset dubbed OPV2V-N. In OPV2V dataset, there is a lack of mask labels for distinguishing between foreground and background views, as well as optical flow labels for supervising the scene flow. For this purpose, we collect more data to bridge the gap between neural field and collaborative perception, leading to the new OPV2V-N datasets. Specifically, we utilize the OneFormer[67] detector to extract the foreground mask labels and mainstream RAFT[68] detector to compute the optical flow between image pairs. Meanwhile, we manually annotate which part of the performance degradation is triggered by camera failure in different scenarios. See more details in the Appendix A.

### 5.2 Quantitative Evaluation

**Benchmark comparison.** The baseline methods include F-Cooper[1], AttFuse[16], DiscoNet[44], V2VNet[37] and CoBEVT[6]. All methods use the same BEV feature encoder based on CVT[69]. To validate the portability of the RCDN, we compare different baseline methods *w.o/w.* RCDN under unpredictable camera failure settings. Table 1 shows that the map-view segmentation performance of different baseline methods *w.o/w.* the proposed RCDN with only one number random noisy camera failure in the testing phase on the OPV2V-N dataset. We see that i) for static part, each baseline method with one camera failure drops about $37.73\%/42.54\%/32.87\%$ (*Avg/Max/Min*) and $52.93\%/61.25\%/44.40\%$ for drivable area and lane, respectively. However, each baseline method *w.* RCDN under the same camera failure situation only decreases about $5.34\%/9.17\%/1.22\%$ and $7.08\%/13.59\%/2.85\%$, respectively. Compared to the *w.o* RCDN baseline methods, RCDN can improve the performance of drivable area and lane for $52.32\%/58.54\%/47.10\%$ and $100.37\%/139.92\%/67.82\%$, respectively; ii) compared to the static part, as we all know, the fusion stage in collaborative perception process needs more effort on the multi-view based BEV feature map to highlight the corresponding dynamic part. Hence, the




(a) Performance on static part Dr. Area (b) Performance on static part Lane (c) Performance on dynamic part Veh.

Figure 3: Comparison of the performance of other baseline methods *w.o/w* the proposed RCDN under the random noisy (failed situation) camera numbers from 0 to 3. RCDN can be ported to other baseline methods and stabilize the performance under different level camera failure situations on OPV2V-N dataset.

Table 2: Ablation Study on OPV2V-N dataset.

| Modules | | Dr. Area | Lanes | Dynamic Veh. |
|---|---|---|---|---|
| Neural Field | Time Model | | | |
| ✗ | ✗ | 24.55 | 10.07 | 30.67 |
| ✔ | ✗ | 24.47 | 11.71 | 41.55 |
| ✔ | ✔ | 27.37 | 10.63 | 46.65 |

Figure 4: Effectiveness of dynamic neural field.

baseline methods' dynamic performance suffers more from camera failure than the static part, causing about a $60.75\%/42.72\%/80.14\%$ performance drop. Nevertheless, RCDN also demonstrates robustness to the dynamic foreground object modeling, with only a $3.31\%/7.58\%/0.47\%$ performance decrease for the dynamic part, improving the *w.o* RCDN baseline methods' performance by $186.57\%/365.19\%/70.01\%$. Meanwhile, as for the communication cost, similar to [44], we only utilize the $\mathbf{C}^{gt}$ labels during the training stage, meaning we leave the communication burden to the training stage and do not introduce extra information during the inference.

**Robust to extremely noisy camera data.** We conduct experiments to validate the performance under the impact of random noisy camera numbers. Figure 3 shows the map-view segmentation performance of the different baselines methods *w.o/w.* the proposed RCDN under varying levels of camera failures situation on OPV2V-N, where the $x$-axis is the expectation of the number of random failed cameras during the inference stage and $y$-axis the segmentation performance. Note that, when the $x$-axis is at 0, it represents standard collaborative perception without any camera failures. We see that i) the proposed RCDN can stabilize all the baseline methods in both static and dynamic part of map-view segmentation performance at all camera failure settings; ii) as for the static part, with the RCDN can maintain the $87.84\%/88.72\%/86.64\%$ Dr. area performance of the standard setting even under three random failed views during the collaboration process, compared with the *w.o.* RCDN only about $47.68\%/57.48\%/37.15\%$. Note that the V2VNet baseline method's performance degrades sharply as the failed camera number increases, however, with RCDN, the V2VNet can settle in a considerable performance even with the failed camera number increases; iii) as for the dynamic part, some baseline methods are crashed even with only one random camera failure situation, *e.g.* DiscoNet only maintains about $19.87\%$ performance of the standard collaborative perception setting, and almost every baseline method is unusable when there are three random camera failures, only about $20.73\%/28.11\%/13.09\%$ of the standard situation. Nevertheless, with the RCDN, we see that all baseline methods still perform well even when three random failed camera situation appear, maintaining the $84.95\%/90.81\%/75.93\%$ dynamic performance of the standard situation.

### 5.3 Qualitative Evaluation

**Visualization of segmentation.** We illustrate the map-view segmentation of other baseline methods *w.o/w.* RCDN and the corresponding repaired camera view in Figure 5. The random camera failure number is one. The orange represents the drivable area, the blue represents the lanes and the teal

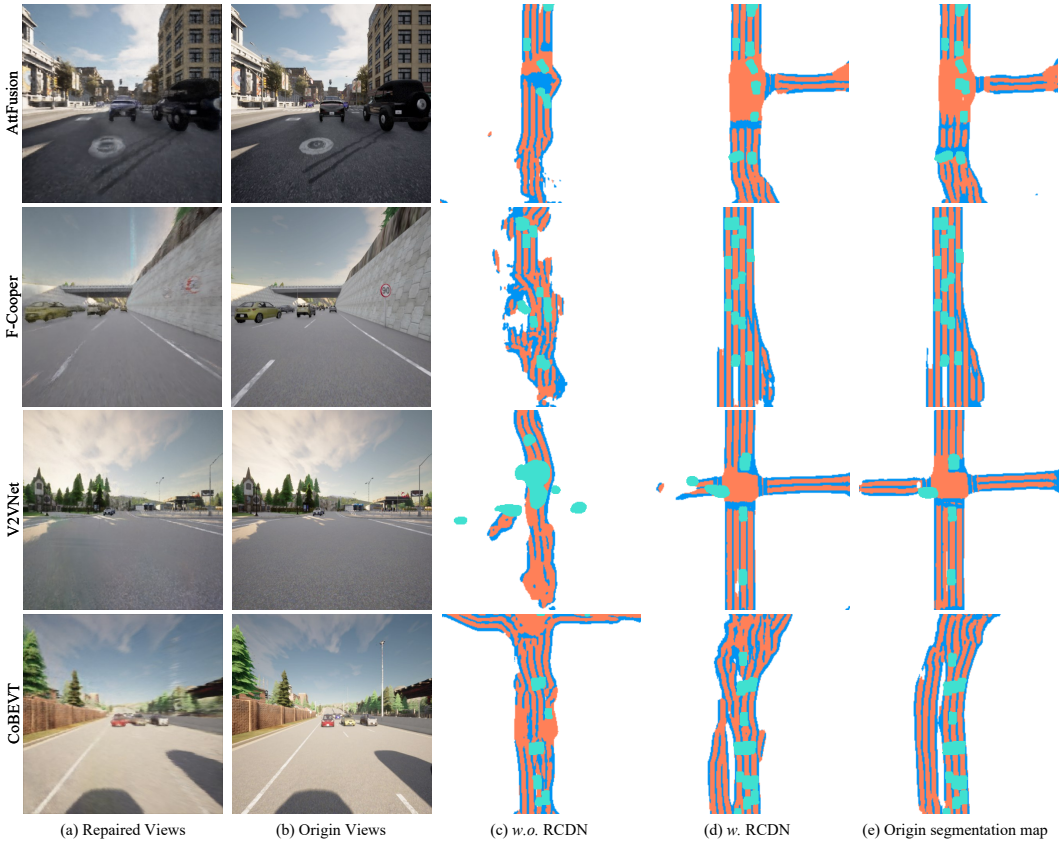



Figure 5: Visualization of different baseline methods *w.* RCDN with one random camera failure.

represents the vehicles. We can see that i) other baseline methods show significant improvement in *w.* RCDN under noisy camera data; ii) V2VNet that collapses with noise camera data can also achieve the same level of performance as the origin data with the help of RCDN.

### 5.4 Ablation Study

**Components analysis** We conduct ablation studies on OPV2V-N with the CoBEVT baseline method. Table 2 assesses the effectiveness of the proposed two field phases. We see that i) only one neural field can recover most static part performance from the noisy camera data; ii) the proposed time model in collaborative dynamic fields can handle the motion blurry caused by the vehicles, shown in Figure 4. Meanwhile, we compare the training efficiency of the proposed RCDN with existing dynamic fields modeling methods[70], as illustrated in Figure 6. Our approach, which leverages explicit grid and geometry feature-based representations, accelerates the training process by approximately $24\times$ compared to the existing implicit MLP-based modeling, while also achieving superior PSNR quality. See more discussions in the Appendix B.2.

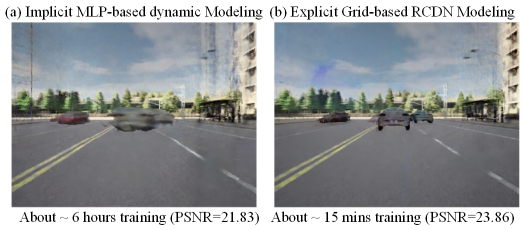



Figure 6: Comparison between existing dynamic field modeling and the proposed RCDN.

**Performance bottlenecks** Regarding the increasing number of agents and cameras, we validated the impact of adding more cameras using the OPV2V-N dataset (corresponding scenario types are T section and midblock respectively) with the CoBEVT baseline. From Table 3, we observe the following: i) With a single overlapping camera view, the proposed method significantly improves baseline performance, and ii) While theoretically, more cameras can provide a larger overlap range, the addition of multiple cameras (depending on their positions) may introduce redundant viewing angles, resulting in less significant performance improvements.

Table 3: Map-view segmentation performance validation about the increasing number of cameras under OPV2V-N datasets with CoBEVT baseline. Note that the failure setting is under one random noisy camera failure in the testing phases. We report IoU for all classes.

| Methods | Metrics | Scene | Failure | Overlap Cameras | | |
|---|---|---|---|---|---|---|
| | | | | +1 | +2 | +3 |
| CoBEVT[6] | Dr. Area | T Section | 23.23 | 26.97 | 26.91 | 27.23 |
| | | Midblock | 23.43 | 38.87 | 38.94 | 39.51 |
| | Dyn. Vehicles | T Section | 18.83 | 40.72 | 41.38 | 42.29 |
| | | Midblock | 16.57 | 45.60 | 48.31 | 49.88 |

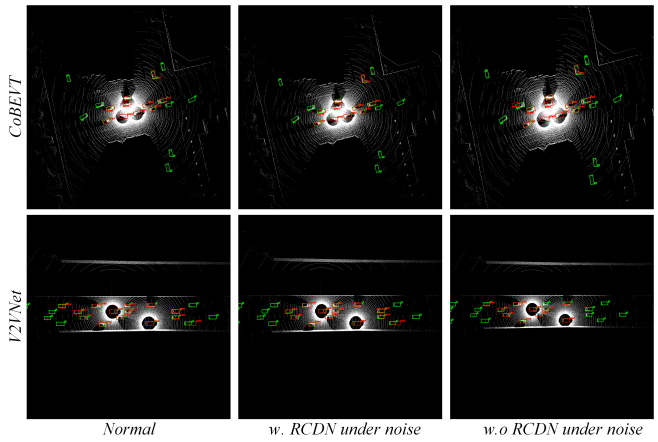



Figure 7: Visualization of proposed RCDN for detection downstream task performance. Note that red and green boxes denote detection results and ground-truth respectively.

Table 4: Detection performance of CoBEVT and V2VNet baseline methods *w.o/w.* the proposed RCDN on OPV2V-N dataset with one random noisy camera failure in the testing phase. We report Average Precision (AP) at Intersection-over-Union (IoU) thresholds of 0.50 and 0.70.

| Methods / Metrics | Normal | |
|---|---|---|
| | AP@0.50(↑) | AP@0.70(↑) |
| CoBEVT[6] | 55.56 | 43.33 |
| V2VNet[37] | 58.77 | 42.42 |
| Methods / Metrics | Failures *w.o/w.* RCDN | |
| | AP@0.50(↑) | AP@0.70(↑) |
| CoBEVT[6] | 46.67/55.56 | 34.57/43.21 |
| V2VNet[37] | 45.45/53.85 | 36.37/38.18 |

**Different downstream tasks** Our proposed RCDN is general to different downstream tasks and is not limited to just BEV segmentation. We focus on BEV segmentation due to its crucial role in autonomous driving, with direct applications to other tasks such as layout mapping, action prediction, route planning, and collision avoidance. Additionally, we have validated RCDN for detection tasks, shown in Figure 7. We replaced the original segmentation header with a detection header in our experiments. Table 4 shows that for CoBEVT, using RCDN improves the metrics of AP@0.50 and AP@0.70 by 19.05% and 24.99%, respectively.

## 6 Conclusion and Limitation

We formulate the camera-insensitivity collaborative perception task, which considers harsh realities of real-world sensors that may cause unpredictable random camera failures during collaborative communication. We further propose RCDN, a robust camera-insensitivity collaborative perception with a novel dynamic feature-based 3D neural modeling. The core idea of RCDN is to construct collaborative neural rendering field representations to recover failed perceptual messages sent by multiple agents. Comprehensive experiments show that RCDN can be portable to other baseline methods and stabilize the performance with a considerable level under all settings and far superior robustness with random camera failures.

**Limitation and future work.** The current work focuses on addressing the camera-insensitivity problem in collaborative perception. It is evident that accurate reconstruction can compensate for the negative impact of noisy camera features on collaborative perception. In the future, we expect more works on exploring real-time collaborative neural field modeling with 3D Gaussian splatting.

## 7 Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2024YFE0211000), in part by the National Natural Science Foundation of China (No. 62372329), in part by Shanghai Scientific Innovation Foundation (No. 23DZ1203400), in part by Tongji-Qomolo Autonomous Driving Commercial Vehicle Joint Lab Project, and in part by Xiaomi Young Talents Program.

## References


[1] Qi Chen, Xu Ma, Sihai Tang, Jingda Guo, Qing Yang, and Song Fu. F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 88–100, 2019.

[2] Runsheng Xu, Yi Guo, Xu Han, Xin Xia, Hao Xiang, and Jiaqi Ma. Opencda: an open cooperative driving automation framework integrated with co-simulation. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1155–1162. IEEE, 2021.

[3] Yiming Li, Dekun Ma, Ziyan An, Zixun Wang, Yiqi Zhong, Siheng Chen, and Chen Feng. V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving. *IEEE Robotics and Automation Letters*, 7(4):10914–10921, 2022.

[4] James Tu, Tsunhsuan Wang, Jingkang Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. Adversarial attacks on multi-agent communication. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7768–7777, 2021.

[5] Jiaxun Cui, Hang Qiu, Dian Chen, Peter Stone, and Yuke Zhu. Coopernaut: End-to-end driving with cooperative perception for networked vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17252–17262, 2022.

[6] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers. 2022.

[7] Hao Xiang, Runsheng Xu, and Jiaqi Ma. Hm-vit: Hetero-modal vehicle-to-vehicle cooperative perception with vision transformer. *arXiv preprint arXiv:2304.10628*, 2023.

[8] Ebtehal Turki Alotaibi, Shahad Saleh Alqefari, and Anis Koubaa. Lsar: Multi-uav collaboration for search and rescue missions. *IEEE Access*, 7:55817–55832, 2019.

[9] Jürgen Scherer, Saeed Yahyanejad, Samira Hayat, Evsen Yanmaz, Torsten Andre, Asif Khan, Vladimir Vukadinovic, Christian Bettstetter, Hermann Hellwagner, and Bernhard Rinner. An autonomous multi-uav system for search and rescue. In *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, page 33–38, New York, NY, USA, 2015.

[10] Yue Hu, Shaoheng Fang, Weidi Xie, and Siheng Chen. Aerial monocular 3d object detection. *IEEE Robotics and Automation Letters*, 8(4):1959–1966, 2023.

[11] Lukas Bernreiter, Shehryar Khattak, Lionel Ott, Roland Siegwart, Marco Hutter, and Cesar Cadena. A framework for collaborative multi-robot mapping using spectral graph wavelets. *The International Journal of Robotics Research*, 0(0):02783649241246847, 0.

[12] Luiz Eugênio Santos Araújo Filho and Cairo Lúcio Nascimento Júnior. Multi-robot autonomous exploration and map merging in unknown environments. In *2022 IEEE International Systems Conference (SysCon)*, pages 1–8, 2022.

[13] Yiming Li, Juexiao Zhang, Dekun Ma, Yue Wang, and Chen Feng. Multi-robot scene completion: Towards task-agnostic collaborative perception. In *6th Annual Conference on Robot Learning*, 2022.

[14] Runsheng Xu, Xin Xia, Jinlong Li, Hanzhao Li, Shuo Zhang, Zhengzhong Tu, Zonglin Meng, Hao Xiang, Xiaoyu Dong, Rui Song, et al. V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13712–13722, 2023.

[15] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, et al. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21361–21370, 2022.

[16] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2583–2589. IEEE, 2022.

[17] Walter Zimmer, Gerhard Arya Wardana, Suren Sritharan, Xingcheng Zhou, Rui Song, and Alois C. Knoll. Tumtraf v2x cooperative perception dataset. In *2024 IEEE/CVF International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE/CVF, 2024.

[18] Runsheng Xu, Jinlong Li, Xiaoyu Dong, Hongkai Yu, and Jiaqi Ma. Bridging the domain gap for multi-agent perception. *arXiv preprint arXiv:2210.08451*, 2022.

[19] Yunsheng Ma, Juanwu Lu, Can Cui, Sicheng Zhao, Xu Cao, Wenqian Ye, and Ziran Wang. Macp: Efficient model adaptation for cooperative perception. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3373–3382, 2024.

[20] Yiming Li, Qi Fang, Jiamu Bai, Siheng Chen, Felix Juefei-Xu, and Chen Feng. Among us: Adversarially robust collaborative perception by consensus. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 186–195, October 2023.

[21] James Tu, Tsunhsuan Wang, Jingkang Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. Adversarial attacks on multi-agent communication. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7748–7757, 2021.

[22] Francesco Secci and Andrea Ceccarelli. On failures of rgb cameras and their effects in autonomous driving applications. In *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*, pages 13–24, 2020.

[23] Kui Ren, Qian Wang, Cong Wang, Zhan Qin, and Xiaodong Lin. The security of autonomous driving: Threats, defenses, and future directions. *Proceedings of the IEEE*, 108(2):357–372, 2020.

[24] Rui Song, Chenwei Liang, Hu Cao, Zhiran Yan, Walter Zimmer, Markus Gross, Andreas Festag, and Alois Knoll. Collaborative semantic occupancy prediction with hybrid feature fusion in connected automated vehicles. 2024.

[25] Xiang Li, Junbo Yin, Wei Li, Chengzhong Xu, Ruigang Yang, and Jianbing Shen. Di-v2x: Learning domain-invariant representation for vehicle-infrastructure collaborative 3d object detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(4):3208–3215, Mar. 2024.

[26] Hao Xiang, Runsheng Xu, Xin Xia, Zhaoliang Zheng, Bolei Zhou, and Jiaqi Ma. V2xp-asg: Generating adversarial scenes for vehicle-to-everything perception. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3584–3591, 2023.

[27] Kaicheng Yu, Tang Tao, Hongwei Xie, Zhiwei Lin, Tingting Liang, Bing Wang, Peng Chen, Dayang Hao, Yongtao Wang, and Xiaodan Liang. Benchmarking the robustness of lidar-camera fusion for 3d object detection. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3188–3198, 2023.

[28] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, pages 1–18. Springer, 2022.

[29] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17830–17839, 2023.

[30] Youngseok Kim, Juyeb Shin, Sanmin Kim, In-Jae Lee, Jun Won Choi, and Dongsuk Kum. Crn: Camera radar net for accurate, robust, efficient 3d perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17615–17626, October 2023.

[31] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1090–1099, June 2022.

[32] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. Bevfusion: A simple and robust lidar-camera fusion framework. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 10421–10434. Curran Associates, Inc., 2022.

[33] Vishwanath A. Sindagi, Yin Zhou, and Oncel Tuzel. Mvx-net: Multimodal voxelnet for 3d object detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7276–7282, 2019.

[34] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 180–191. PMLR, 08–11 Nov 2022.

[35] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.

[36] Maria Christopoulou, Sokratis Barmpounakis, Harilaos Koumaras, and Alexandros Kaloxylos. Artificial intelligence and machine learning as key enablers for v2x communications: A comprehensive survey. *Vehicular Communications*, 39:100569, 2023.

[37] Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, and Raquel Urtasun. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 605–621, 2020.

[38] Yifan Lu, Yue Hu, Yiqi Zhong, Dequan Wang, Siheng Chen, and Yanfeng Wang. An extensible framework for open heterogeneous collaborative perception. In *The Twelfth International Conference on Learning Representations*, 2024.

[39] Yue Hu, Yifan Lu, Runsheng Xu, Weidi Xie, Siheng Chen, and Yanfeng Wang. Collaboration helps camera overtake lidar in 3d detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9243–9252, 2023.

[40] Yen-Cheng Liu, Junjiao Tian, Chih-Yao Ma, Nathan Glaser, Chia-Wen Kuo, and Zsolt Kira. Who2com: Collaborative perception via learnable handshake communication. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6876–6883, 2020.

[41] Yen-Cheng Liu, Junjiao Tian, Nathaniel Glaser, and Zsolt Kira. When2com: multi-agent perception via communication graph grouping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4106–4115, 2020.

[42] Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. *Advances in neural information processing systems*, 35:4874–4886, 2022.

[43] Tianhang Wang, Guang Chen, Kai Chen, Zhengfa Liu, Bo Zhang, Alois Knoll, and Changjun Jiang. Umc: A unified bandwidth-efficient and multi-resolution based collaborative perception framework. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8153–8162, 2023.

[44] Yiming Li, Shunli Ren, Pengxiang Wu, Siheng Chen, Chen Feng, and Wenjun Zhang. Learning distilled collaboration graph for multi-agent perception. *Advances in Neural Information Processing Systems*, 34:29541–29552, 2021.

[45] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. *ArXiv*, abs/2203.10638, 2022.

[46] Nicholas Vadivelu, Mengye Ren, James Tu, Jingkang Wang, and Raquel Urtasun. Learning to communicate and correct pose errors. In *4th Conference on Robot Learning (CoRL)*, 2020.

[47] Yifan Lu, Quanhao Li, Baoan Liu, Mehrdad Dianati, Chen Feng, Siheng Chen, and Yanfeng Wang. Robust collaborative 3d object detection in presence of pose errors. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4812–4818. IEEE, 2023.

[48] Zixing Lei, Shunli Ren, Yue Hu, Wenjun Zhang, and Siheng Chen. Latency-aware collaborative perception. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 316–332. Springer, 2022.

[49] Sizhe Wei, Yuxi Wei, Yue Hu, Yifan Lu, Yiqi Zhong, Siheng Chen, and Ya Zhang. Asynchrony-robust collaborative perception via bird's eye view flow. In *Advances in Neural Information Processing Systems*, 2023.

[50] Dian Chen and Philipp Krähenbühl. Learning from all vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17222–17231, 2022.

[51] Ruizhao Zhu, Peng Huang, Eshed Ohn-Bar, and Venkatesh Saligrama. Learning to drive anywhere. In *7th Annual Conference on Robot Learning*, 2023.

[52] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[53] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.

[54] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023.

[55] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19774–19783, 2023.

[56] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022.

[57] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5459–5469, 2022.

[58] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensoir: Tensorial inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2023.

[59] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[60] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[61] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

[62] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12932–12942, 2022.

[63] Fan Lu, Yan Xu, Guang Chen, Hongsheng Li, Kwan-Yee Lin, and Changjun Jiang. Urban radiance field representation with deformable neural mesh primitives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 465–476, 2023.

[64] Fan Lu, Kwan-Yee Lin, Yan Xu, Hongsheng Li, Guang Chen, and Changjun Jiang. Urban architect: Steerable 3d urban scene generation with layout prior. *arXiv preprint arXiv:2404.06780*, 2024.

[65] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[66] Robert A Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *ACM Siggraph Computer Graphics*, 22(4):65–74, 1988.

[67] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. OneFormer: One Transformer to Rule Universal Image Segmentation. 2023.

[68] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[69] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13750–13759, 2022.

[70] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: This paper does discuss the limitations of the work performed by the authors.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: This paper fully discloses all the information needed to reproduce the main experimental results of the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: This paper provides open access to the data and code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: This paper specify all the training and test details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: This paper reports appropriate information about the statistical significance of the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This paper provides sufficient information on the computer resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research conducted in the paper conform, in evrery respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed. No harm technical paper.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We mentioned creators or original owners of assets and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: This paper introduces new assets well documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.