2023 Design, Automation & Test in Europe Conference (DATE 2023)

# A Safety-Guaranteed Framework for Neural-Network-Based Planners in Connected Vehicles under Communication Disturbance

Kevin Kai-Chun Chang[1], Xiangguo Liu[2], Chung-Wei Lin[1], Chao Huang[3], Qi Zhu[2]
[1]National Taiwan University, [2]Northwestern University, [3]University of Liverpool
kevin.kaichun.chang@gmail.com, xg.liu@u.northwestern.edu, cwlin@csie.ntu.edu.tw,
chao.huang2@liverpool.ac.uk, qzhu@northwestern.edu

***Abstract*—Neural-network-based (NN-based) planners have been increasingly used to enhance the performance of planning for autonomous vehicles. However, it is often difficult for NN-based planners to balance efficiency and safety in complicated scenarios, especially under real-world communication disturbance. To tackle this challenge, we present a safety-guaranteed framework for NN-based planners in connected vehicle environments with communication disturbance. Given any NN-based planner with no safety-guarantee, the framework generates a robust compound planner embedding the NN-based planner to ensure overall system safety. Moreover, with the aid of an information filter for imperfect communication and an aggressive approach for the estimation of the unsafe set, the compound planner could achieve similar or better efficiency than the given NN-based planner. A comprehensive case study of unprotected left turn and extensive simulations demonstrate the effectiveness of our framework.**

***Index Terms*—neural-network-based planning, safety guarantee, connected vehicles, communication disturbance**

## I. Introduction

Learning-based methods have shown great promise in the planning and control of connected and autonomous vehicles [1]–[5], and neural-network-based (NN-based) planners are becoming increasingly popular [6], [7]. In comparison with traditional model-based planners, NN-based approaches could effectively capture system properties and enhance average vehicle planning performance/efficiency under various scenarios. However, a major challenge of NN-based planners is to balance efficiency and safety. To fully ensure system safety, some NN-based planners could become overly conservative and thus significantly sacrifice efficiency [8]. On the other hand, over-aggressive NN-based planners may violate safety constraints when trying to enhance efficiency. Balancing the two objectives with the safety verification of NN-based planners is very challenging and time-consuming, especially under complicated scenarios [6], [9].

In connected vehicle environments, NN-based planners could leverage the messages from other vehicles to improve their own sensor-based estimation. However, previous works in connected vehicles often assume perfect communication [1]–[3], [10], [11], while communication disturbance [12] including message transmission delays [13] and drops is common in real-world connected vehicle environments. Such imperfect communication makes it much harder to leverage the messages from other vehicles in NN-based planning. Besides, the inaccuracies of sensor measurements [14] are often neglected as well [11].

To overcome the above challenges, we propose a safety-guaranteed framework for NN-based planners in connected vehicle environments with communication disturbance and sensor inaccuracies. First, given any NN-based planner, the framework generates a *compound planner* embedding the NN-based planner. Most of the time, the compound planner adopts the planning decision from the NN-based planner; however, when a *runtime monitor* assesses that safety constraints are about to be violated, the compound planner switches to an *emergency planner* to ensure the overall system safety. Second, two techniques, *information filter* and *aggressive unsafe set estimation*, are proposed to further enhance system efficiency. The information filter fuses the reachability analysis and a Kalman Filter to extract information from imperfect communication and sensors, and the aggressive unsafe set estimation generates a reduced unsafe set for the embedded NN-based planner. As a result, both safety and efficiency are taken into account.

The main contributions of this work are summarized below:

- We propose a novel safety-guaranteed framework that could be applied to any NN-based planner in connected vehicle environments under communication disturbance.
- We design the information filter and the aggressive unsafe set estimation methods, which could effectively reduce the size of the unsafe set and thus enhance the efficiency.
- We conduct a comprehensive case study of the unprotected left turn scenario. The experiment results demonstrate the safety and efficiency improvement from our framework, in comparison with the pure NN-based planners.

The remainder of this paper is organized as follows: Section II defines the system model and formulates our problem. Section III details the design of our compound planner. Section IV illustrates the case study of unprotected left turn. Section V shows experimental results. Finally, Section VI concludes the paper.

## II. System Model and Problem Formulation

### A. System Model

In this paper, we consider a general and discrete system of connected vehicles described as follows.

**Vehicle.** A vehicle $C_i$ can be described as follows (here we adopt a one-dimensional system as a representative):

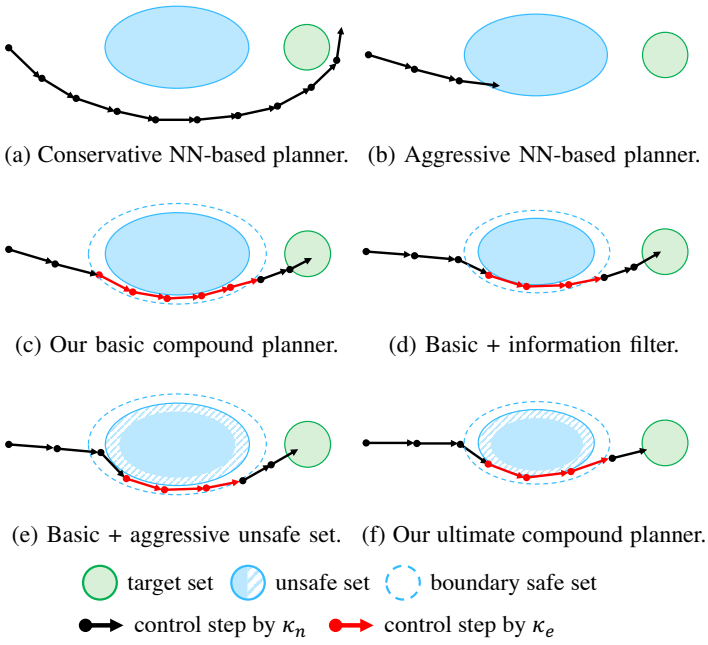
$$\begin{bmatrix} p_i(t+\Delta t_c) \\ v_i(t+\Delta t_c) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t_c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_i(t) \\ v_i(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta t_c^2 \\ \Delta t_c \end{bmatrix} a_i(t), t \geq 0,$$

where $\Delta t_c$ is the control time step, $p_i(t)$, $v_i(t)$, and $a_i(t)$ are the position, velocity, and acceleration of $C_i$, respectively.

**Ego vehicle.** The ego vehicle $C_0$ is the vehicle controlled by our planner. In other words, our planner determines the acceleration, $a_0(t)$, of the ego vehicle at each timestamp $t$.

**Unsafe set.** The unsafe set $X_u$ includes all states causing violations of safety constraints. For example, if the ego vehicle $C_0$ and another vehicle $C_i$ are on the same lane, $C_0$ must keep a distance gap with $C_i$ to avoid collision. Therefore, the unsafe set could be defined as $X_u = \{x(t) \mid |p_0(t) - p_i(t)| < p_{gap}\}$, where $x(t)$ is the system state (including positions and velocities of all the vehicles) at $t$, and $p_{gap}$ is the minimum distance gap to ensure safety.

978-3-9819263-7-8/DATE23/© 2023 EDAA




(a) Conservative NN-based planner. (b) Aggressive NN-based planner.

(c) Our basic compound planner. (d) Basic + information filter.

(e) Basic + aggressive unsafe set. (f) Our ultimate compound planner.

Fig. 1: Schematic of planners' behavior. Planners from (c) to (f) are the contributions of this paper.

**Target set.** The target set $X_t$ consists of the destination states of the ego vehicle. For example, consider a lane-changing scenario. If the ego vehicle aims to change from Lane A to Lane B, the target set can be defined as $X_t = \{x(t) \mid C_0 \text{ is on Lane B.}\}$.

**Message.** We assume that every $\Delta t_m$ seconds, the ego vehicle $C_0$ receives messages, $p_{i,m}(t)$, $v_{i,m}(t)$, and $a_{i,m}(t)$ $(i = 1, 2, \ldots, n-1)$, from other vehicles. The message content is accurate (i.e., $p_{i,m}(t) = p_i(t)$, $v_{i,m}(t) = v_i(t)$, and $a_{i,m}(t) = a_i(t)$); however, the message may delay with $\Delta t_d$ (i.e., $C_0$ receives the message of timestamp $t$ at $t + \Delta t_d$), or even drop (i.e., $\Delta t_d \to \infty$). Note that the system model could also be applied to the circumstance where vehicles are not connected (i.e., messages always drop).

**Sensor.** In addition to messages from other vehicles, another way for the ego vehicle to estimate the information of other vehicles is via its own onboard sensors. Every $\Delta t_s$ seconds, the ego vehicle $C_0$ senses the information, $p_{i,s}(t)$, $v_{i,s}(t)$, and $a_{i,s}(t)$ $(i = 1, 2, \ldots, n-1)$, of other vehicles. In contrast to the messages, we assume that $C_0$ always obtains the sensor-measured values without delay (i.e., the computation delay is negligible); however, such measurement is inaccurate. For example, $p_{i,s}(t)$ could be any number between $p_i(t) - \delta_p$ and $p_i(t) + \delta_p$, where $\delta_p$ is the uncertainty of the sensor in detecting the position. (We assume a uniform distribution in $[p_i(t) - \delta_p, p_i(t) + \delta_p]$). Besides, $\delta_v$ and $\delta_a$ are the uncertainty in velocity and acceleration, respectively.

**Planner.** A planner $\kappa_j$ determines the acceleration of the ego vehicle $C_0$ at every timestamp $t$ based on the current system state, i.e., $a_0(t) = \kappa_j(x(t))$.

**Evaluation.** To evaluate the efficiency and safety guarantee of a planner $\kappa_j$, we define an evaluation function $\eta$ as follows:

$$\eta(\kappa_j) = \begin{cases} -1, & \text{if } \exists t_k, \text{ s.t., } x(t_k) \in X_u \text{ and } \forall t < t_k, x(t) \notin X_t; \\ \frac{1}{t_r}, & \text{else if } \exists t_r, \text{ s.t., } x(t_r) \in X_t \text{ and } \forall t < t_r, x(t) \notin X_t; \\ 0, & \text{otherwise.} \end{cases}$$

A larger value of $\eta(\kappa_j)$ indicates a better performance/efficiency of $\kappa_j$. As we can see, the evaluation function views safety as the top priority, and thus a violation of safety constraints causes $\eta(\kappa_j) = -1$. If $\kappa_j$ could make the vehicle reach the target set safely, its efficiency is further evaluated by the reaching time $t_r$.

### B. Problem Formulation

Given a system of connected vehicles defined in Section II-A and a neural-network-based planner $\kappa_n$, our framework aims to find a compound planner $\kappa_c$ which embeds $\kappa_n$, such that $\kappa_c$ could achieve similar or better efficiency than $\kappa_n$. Moreover, $\kappa_c$ should always guarantee system safety, even under imperfect communication (i.e., with message delays and drops).

Formally, $\kappa_c$ needs to satisfy:

$$\eta(\kappa_c) \geq \eta(\kappa_n) \text{ and } \eta(\kappa_c) \geq 0. \tag{1}$$

## III. Our Framework

### A. Overview of Compound Planner Design

As aforementioned, it is difficult for pure NN-based planners $\kappa_n$ to balance both safety and efficiency under communication disturbance and complicated scenarios. A conservative NN-based planner, as shown in Figure 1a, often detours to avoid the unsafe set, and may even miss the target set. On the other hand, as shown in Figure 1b, an aggressive planner is liable to enter the unsafe set on its way to the target set.

To achieve a better balance between safety and efficiency, we introduce the runtime monitor and the emergency planner $\kappa_e$ to form a compound planner with $\kappa_n$. They play as the "last line of defense" for the system safety. That is, if the ego vehicle is about to run into the unsafe set, the runtime monitor will select the emergency planner $\kappa_e$ to ensure the ego vehicle stays in the safe set. Otherwise, the runtime monitor will just select the NN-based planner $\kappa_n$ (the selection criteria is detailed in Section III-C). Note that the emergency planner is only selected when safety constraints are about to be violated; thus, the efficiency degradation due to $\kappa_e$ is minimized. As illustrated in Figure 1c, the basic compound planner could guarantee system safety while maintaining efficiency.

To further enhance the system efficiency, we observe that the planner's behavior is greatly influenced by the size of the unsafe set. When the unsafe set is too large, $\kappa_n$ tends to control the ego vehicle in a more conservative manner to avoid violation of safety. Besides, a large unsafe set also implies $\kappa_e$ is likely to be selected and hence damaging to the efficiency. On the other hand, if the unsafe set is overly underestimated, $\kappa_n$ may be misled and thus incur $\kappa_e$ as well. Therefore, an adequate and precise estimated unsafe set is desirable.

We employ two techniques to reduce the size of the estimated unsafe set moderately. First, we perform reachability analysis on the delayed messages and adopt the Kalman Filter to extract accurate information from uncertain sensor measurements (the implementation details are described in Section III-B). Through the information processing, the unsafe set could be estimated more precisely, as shown in Figure 1d. Second, as the runtime monitor and the emergency planner already guarantee the overall system safety, we do not need to consider the whole unsafe set when manipulating $\kappa_n$. Instead, we feed an underestimated unsafe set, $X_{u,aggr}$ (called the aggressive unsafe set), which includes only the "core" part of the unsafe set, into $\kappa_n$. It is reasonable since many extreme states in the original $X_u$ seldom occur. Therefore, discarding these states could avoid over-conservative planning. As illustrated in Figure 1e, $\kappa_n$ only adopts the solid part while neglecting the slashed part, resulting in more aggressive planning.

Eventually, combining the runtime monitor and the emergency planner with the two techniques, our compound planner could not only guarantee system safety but also improve efficiency compared to the pure NN-based planner, as shown in Figure 1f.

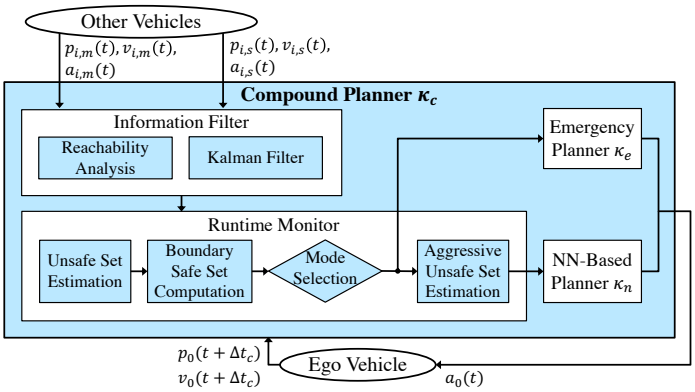



Fig. 2: The architecture of the compound planner $\kappa_c$.

In summary, Figure 2 illustrates the full design of our compound planner $\kappa_c$. First, the received messages and sensor-measured values are passed through the information filter. Then, the runtime monitor could compute a precise unsafe set based on the processed information. After that, it selects either the emergency planner or the NN-based planner accordingly to determine the next control input (i.e., $a_0(t)$) for the ego vehicle. If the NN-based planner is selected, the aggressive unsafe set is computed for the NN-based planner's planning.

### B. Information Filter

As aforementioned, we apply reachability analysis and Kalman Filter on delayed messages and inaccurate sensor measured values, respectively. Afterward, the estimations are joined together and passed to the runtime monitor. For example, if the possible position of $C_i$ is estimated as any value in $[p_{i,1}(t), p_{i,2}(t)]$ via reachability analysis, and $[p_{i,3}(t), p_{i,4}(t)]$ via Kalman Filter, then the joined estimation of $C_i$'s position is $[\max(p_{i,1}(t), p_{i,3}(t)), \min(p_{i,2}(t), p_{i,4}(t))]$.

**Kalman Filter.** We adopt the Kalman Filter [15], [16], one of the most common and effective filters, to recover information from uncertain sensor measurements. As Figure 3 illustrates, the Kalman Filter estimates the real current state based on historical and current measurements. In each sensing period, the Kalman Gain, $K_i(t)$, is computed. Then, the estimate of the system state (*resp.* covariance), $\hat{x}_i(t,t)$ (*resp.* $P(t,t)$), is updated based on the Kalman Gain, the new measured values $p_{i,s}(t)$ and $v_{i,s}(t)$, and the prediction from the previous iteration, $\hat{x}_i(t, t-\Delta t_s)$ (*resp.* $P_i(t, t-\Delta t_s)$). After that, the prediction of the next state (*resp.* covariance), $\hat{x}_i(t+\Delta t_s, t)$ (*resp.* $P_i(t+\Delta t_s, t)$), is extrapolated according to the measured value and uncertainty of the control input. The corresponding equations for the connected vehicle system defined in Section II-A could be deduced as follows [16]:

$$\begin{aligned}
&\hat{x}_i(t+\Delta t_s, t) = F\hat{x}_i(t,t) + Ga_{i,s}(t),\\
&P_i(t+\Delta t_s, t) = FP_i(t,t)F^T + Q,\\
&K_i(t) = P_i(t, t-\Delta t_s)(P_i(t, t-\Delta t_s) + R)^{-1},\\
&\hat{x}_i(t,t) = \hat{x}_i(t, t-\Delta t_s) + K_i(t)(x_{i,s}(t) - \hat{x}_i(t, t-\Delta t_s)),\\
&P(t,t) = (I - K_i(t))P_i(t, t-\Delta t_s)(I - K_i(t))^T + K_i(t)RK_i(t)^T,
\end{aligned}$$

where

$$F = \begin{bmatrix} 1 & \Delta t_s \\ 0 & 1 \end{bmatrix}, G = \begin{bmatrix} 0.5\Delta t_s^2 \\ \Delta t_s \end{bmatrix}, Q = \begin{bmatrix} 0.25\Delta t_s^4 & 0.5\Delta t_s^3 \\ 0.5\Delta t_s^3 & \Delta t_s^2 \end{bmatrix} \frac{\delta_a^2}{3},$$
$$R = \begin{bmatrix} \frac{\delta_p^2}{3} & 0 \\ 0 & \frac{\delta_v^2}{3} \end{bmatrix}, x_{i,s}(t) = \begin{bmatrix} p_{i,s}(t) \\ v_{i,s}(t) \end{bmatrix}, I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

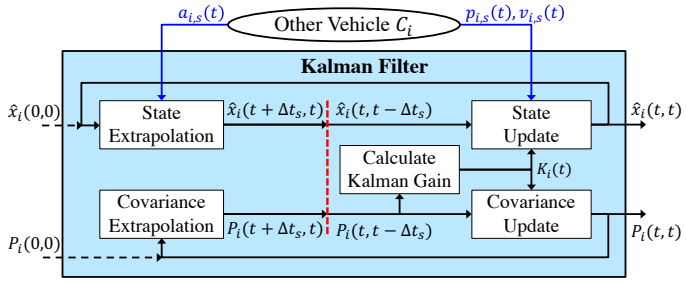



Fig. 3: The flow of Kalman Filter [16]. The red dashed line is the boundary of a sensing period. $\hat{x}_i(0,0)$ and $\hat{P}_i(0,0)$ are the initial values of $C_i$'s states and covariance, respectively.

To enhance the predictability of the Kalman Filter, we modify the traditional design and further incorporate the message information into the filter. In each transmission period $t$, the extrapolated state $\hat{x}_i(t, t-\Delta t_s)$ and the covariance $P_i(t, t-\Delta t_s)$ are stored in the memory. Then, every time a message recording the states of $C_i$ at time $t_k$ arrives, $\hat{x}_i(t_k, t_k - \Delta t_s)$ and $P_i(t_k, t_k - \Delta t_s)$ are restored, and the filter renews the estimations from $t_k$ to the current timestamp based on $x_{i,m}(t_k)$ and $a_{i,m}(t_k)$.

**Reachability analysis.** To deal with message delay or drop, we perform reachability analysis on the latest received message information. For example, assume that the current time is $t$ and the latest message records the states of $C_i$ at time $t_k$, i.e., $p_i(t_k)$ and $v_i(t_k)$. Then, the current position of $C_i$ could be any value between $p_{i,min}(t)$ and $p_{i,max}(t)$, where

$$p_{i,max}(t) = \begin{cases} p_i(t_k) + v_i(t_k)(t-t_k) + \frac{1}{2}a_{i,max}(t-t_k)^2, \\ \qquad \text{if } v_i(t_k) + a_{i,max}(t-t_k) \le v_{i,max}; \\ p_i(t_k) + v_{i,max}(t-t_k) - \frac{(v_{i,max} - v_i(t_k))^2}{2a_{i,max}}, \\ \qquad \text{otherwise}, \end{cases} \tag{2}$$

where $v_{i,max}$ and $a_{i,max}$ are the maximum velocity and acceleration of $C_i$, respectively, and $p_{i,min}(t)$ could be computed in a similar way. Then, we can estimate the unsafe set based on the interval $[p_{i,min}(t), p_{i,max}(t)]$ to ensure system safety.

### C. Runtime Monitor

To formally describe the functionality of the runtime monitor, we should define the *boundary safe set* $X_b$ first:

$$X_b = \{x(t) \in X_s \mid \exists a_0(t), \ldots, a_{n-1}(t) \text{ s.t. } x(t+\Delta t_c) \in X_u\}, \tag{3}$$

where $X_s$ is the safe set. The boundary safe set includes all the states that are only one step away from the unsafe set.

In each control step, the runtime monitor estimates the unsafe sets $X_u$ based on the filtered information of other vehicles. After that, the boundary safe set is computed by Equation (3). Then, the runtime monitor selects the emergency planner **if and only if** the current state $x(t)$ is in the boundary safe set.

If the NN-based planner is selected, i.e., the ego vehicle will not enter the unsafe set in the next step under any valid control input, the runtime monitor further computes the aggressive unsafe set for the NN-based planner. For example, as vehicles rarely change their velocities drastically, we may replace $a_{i,max}$ in Equation (2) with a much smaller value (e.g., $a_i(t)$) to estimate $p_{i,max}(t)$. In this way, the neural-network-based planner could control the ego vehicle more aggressively, hence the improvements in efficiency without making sacrifices for safety.

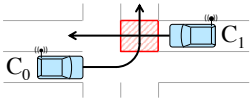



Fig. 4: The unprotected left turn scenario.

### D. Emergency Planner

As for the emergency planner $\kappa_e$, it must satisfy the following equation:

$$\forall x(t) \in X_b, x(t+\Delta t_c) \in X_s \text{ under the control of } \kappa_e. \quad (4)$$

That is, the emergency planner always keeps the ego vehicle within the safe set. An example of emergency planner design is demonstrated in Section IV.

### E. Analysis of Compound Planner

In this subsection, we verify if the compound planner satisfies Equation (1).

**Safety: $\eta(\kappa_c) \geq 0$.** Based on proof by contradiction, assume the ego vehicle enters the unsafe set at time $t_k$. Then, according to Equation (3), the ego vehicle belongs to the boundary safe set at some time $t_l$ with $t_l < t_k$. However, the emergency planner must take over the control at $t_l$ and keep the ego vehicle staying in the safe set, according to Equation (4). Therefore, the ego vehicle will never enter the unsafe set and thus $\eta(\kappa_c) \geq 0$.

The above analysis also indicates as long as the design of the emergency planner satisfies Equation (4), the ego vehicle will never enter the unsafe set. Therefore, it does not require extra resources for safety verification during runtime.

**Efficiency: $\eta(\kappa_c) \geq \eta(\kappa_n)$.** Consider the following two cases:

1) The ego vehicle has never entered the boundary safe set before reaching the target set.
2) The ego vehicle once entered the boundary safe set.

In the first case, the runtime monitor always selects $\kappa_n$ to control during the process. Therefore, the efficiency of $\kappa_c$ should be at least the same as $\kappa_n$, thus $\eta(\kappa_c) \geq \eta(\kappa_n)$. For the second case, the emergency planner $\kappa_e$ has been selected. Although $\kappa_e$ may lower the efficiency, the two techniques, information filter and aggressive unsafe set estimation, could compensate for the sacrifice. Moreover, under the control of a pure NN-based planner, the ego vehicle is liable to fall into the unsafe set once it runs into the boundary safe set, and thus it is very likely that $\eta(\kappa_n) < 0$. On the other hand, the compound planner always ensures the safety of the ego vehicle, hence $\eta(\kappa_c) \geq 0$. As a result, $\eta(\kappa_c) \geq \eta(\kappa_n)$ holds for most of the time.

## IV. Case Study: Unprotected Left Turn

Figure 4 illustrates the unprotected left turn scenario of our case study. The ego vehicle, $C_0$, aims to turn left, while there is another oncoming vehicle $C_1$ in the opposite direction. A collision occurs if the two vehicles are in the unsafe area (i.e., the red rectangle) at the same time. To simplify the system model, we assume that the paths of the two vehicles are fixed. Therefore, it becomes a one-dimensional system, and the planner only needs to determine the acceleration of $C_0$ at each timestamp.

Inspired by the work in [6] on unprotected left turn, we define the input variables of the NN-based planner $\kappa_n$ as $t$, $p_0(t)$, $v_0(t)$, $\tau_{1,min}(t)$, and $\tau_{1,max}(t)$, where $[\tau_{1,min}(t), \tau_{1,max}(t)]$ is the possible time window for $C_1$ to pass the unsafe area. Consequently, to apply our framework to the unprotected left turn scenario, we could define the unsafe set $X_u$ using these five variables. Then, we could compute the boundary safe set, design the emergency planner, and estimate the aggressive unsafe set accordingly.

**Unsafe set estimation.** To ensure system safety, $C_0$ and $C_1$ should not appear in the unsafe area simultaneously. Thus, the unsafe set $X_u$ contains all the states that may result in such situations. To formally compute $X_u$, we first define the *slack* $s(t)$ and the *projected passing time interval* $[\tau_{0,min}(t), \tau_{0,max}(t)]$:

$$s(t) = \begin{cases} p_f - d_b - p_0(t), & \text{if } p_0(t) \leq p_f; \\ p_0(t) - p_b, & \text{else if } p_0(t) \leq p_b; \\ \infty, & \text{otherwise,} \end{cases} \quad (5)$$

$$[\tau_{0,min}(t), \tau_{0,max}(t)] = \begin{cases} [t + \frac{p_f - p_0(t)}{v_0(t)}, t + \frac{p_b - p_0(t)}{v_0(t)}], & \text{if } p_0(t) \leq p_f; \\ [t, t + \frac{p_b - p_0(t)}{v_0(t)}], & \text{else if } p_0(t) \leq p_b; \\ [0, 0] & \text{otherwise,} \end{cases}$$

where $p_f$ is the front line of the unsafe area, $p_b$ is the back line of the unsafe area, and $d_b = -0.5v_0(t)^2/a_{0,min}(t)$ is the braking distance of $C_0$. A nonnegative slack indicates $C_0$ is able to stop before the front line; on the other hand, a negative slack implies $C_0$ must run into the unsafe area before it stops. $[\tau_{0,min}(t), \tau_{0,max}(t)]$ represents the projected passing time window of $C_0$ to the unsafe area under current velocity. Then, the unsafe set $X_u$ could be defined as:

$$X_u = \{(t, p_0(t), v_0(t), \tau_{1,min}(t), \tau_{1,max}(t)) \mid s(t) < 0 \text{ and } [\tau_{0,min}(t), \tau_{0,max}(t)] \cap [\tau_{1,min}(t), \tau_{1,max}(t)] \neq \emptyset\}. \quad (6)$$

That is, the projected passing time windows of $C_0$ and $C_1$ intersect, but $C_0$ is unable to stop before the unsafe area. Thus, collisions may occur.

Lastly, note that $C_0$ only obtains the position, velocity, and acceleration of $C_1$, the runtime monitor needs to estimate $\tau_{1,min}(t)$ and $\tau_{1,max}(t)$ on its own:

$$\tau_{1,min}(t) = \begin{cases} \frac{v_{1,max} - v_1(t)}{a_{1,max}} + \frac{p_f - p_1(t) - d_{th}}{v_{1,max}}, & \text{if } p_f - p_1(t) > d_{th}; \\ \frac{-v_1(t) + \sqrt{v_1(t)^2 + a_{1,max}(p_f - p_1(t))}}{a_{1,max}}, & \text{otherwise,} \end{cases} \quad (7)$$

where $d_{th} = \frac{v_{1,max}^2 - v_1(t)^2}{2a_{1,max}}$ is just for simplification. $\tau_{1,max}(t)$ can be estimated similarly based on $v_{1,min}$, $a_{1,min}$, and $p_b$. For convenience, here we assume that $C_0$ obtains the exact values of $p_1(t)$ and $v_1(t)$ (i.e., perfect communication). When there exist message delays, the reachability analysis like Equation (2) needs to be performed first. Furthermore, the uncertainties $\delta_p$ and $\delta_v$ should be taken into consideration as well.

**Boundary safe set computation.** According to Equation (3) and Equation (6), the boundary safe set at timestamp $t$ for the unprotected left turn scenario are the states that may lead to a negative slack value at $t + \Delta t_c$. The minimum possible slack at the next timestamp $t + \Delta t_c$ is:

$$\begin{aligned} s(t+\Delta t_c) &= p_f - d_b(t+\Delta t_c) - p_0(t+\Delta t_c) \\ &\geq p_f - \frac{(v_0(t) + a_{0,max}\Delta t_c)^2}{2a_{0,min}} - (p_0(t) + v_0(t) + \frac{a_{0,max}\Delta t_c^2}{2}) \\ &= s(t) - (v_0(t)\Delta t_c + \frac{1}{2}a_{0,max}\Delta t_c^2)(1 - \frac{a_{0,max}}{a_{0,min}}), \end{aligned}$$

which should be nonnegative. Thus, combining with Equation (5), the boundary safe set is:

$$X_b = \{(t, p_0(t), v_0(t), \tau_{1,min}(t), \tau_{1,max}(t)) \mid 0 \leq s(t) < (v_0(t)\Delta t_c + \frac{1}{2}a_{0,max}\Delta t_c^2)(1 - \frac{a_{0,max}}{a_{0,min}}) \text{ and } [\tau_{0,min}(t), \tau_{0,max}(t)] \cap [\tau_{1,min}(t), \tau_{1,max}(t)] \neq \emptyset\}.$$

**Aggressive unsafe set estimation.** Most of the time, other vehicles would neither accelerate with the extreme values nor travel at the maximum (or minimum) velocity. Therefore, instead of using the physical limits $a_{1,max}$ and $v_{1,max}$ to estimate $\tau_{1,min}(t)$ as Equation (7), we could adopt $a_{1,est} = min(a_1(t)+a_{buf}, a_{1,max})$ and $v_{1,est} = min(v_1(t)+v_{buf}, v_{1,max})$, respectively, where $a_{buf}$ and $v_{buf}$ are user-defined buffers. It is reasonable since vehicles usually change their velocity within a limited range. In this way, we could estimate a more compact passing time window and hence a smaller unsafe set. Moreover, the aggressive passing time window is still close to the real passing time; therefore, the ego vehicle would not fall into the emergency planner easily.

$$\tau_{1,min}(t) = \begin{cases} \frac{v_{1,est}-v_1(t)}{a_{1,est}} + \frac{p_f-p_1(t)-d_{th}}{v_{1,est}}, & \text{if } p_f - p_1(t) > d_{th}; \\ \frac{-v_1(t)+\sqrt{v_1(t)^2+a_{1,est}(p_f-p_1(t))}}{a_{1,est}}, & \text{otherwise,} \end{cases} \tag{8}$$

where $d_{th} = \frac{v_{1,est}^2 - v_1(t)^2}{2a_{1,est}}$ is just for simplification. $\tau_{1,min}(t)$ can be estimated similarly based on $max(v_1(t) - v_{buf}, v_{1,min})$, $max(a_1(t) - a_{buf}, a_{1,min})$, and $p_b$.

**Emergency planner.** According to Equation (4), $\kappa_e$ should always keep $C_0$ away from $X_u$. The acceleration generated by $\kappa_e$ could be set as:

$$\kappa_e(x(t)) = a_0(t) = \begin{cases} -\frac{v_0(t)^2}{2(p_f-p_0(t))}, & \text{if } p_0(t) \le p_f; \\ a_{0,max}, & \text{otherwise.} \end{cases}$$

If $C_0$ has not yet entered the unsafe area, $\kappa_e$ will make it stop before the unsafe area with the least required braking force. Otherwise, $\kappa_e$ will let $C_0$ escape from the unsafe area as soon as possible.

## V. Experimental Results

Our planner is implemented in the C++ programming language, and the experiments are conducted on a Linux workstation with 3.7 GHz CPU and 192 GB RAM.

### A. Effectiveness of the Proposed Framework

As aforementioned, some NN-based planners are overly conservative while some are too aggressive. Hence, based on the learning methods in [6], we implement two NN-based planners, $\kappa_{n,cons}$ and $\kappa_{n,aggr}$, to represent the two designs. Then, we compare these pure NN-based planners with the corresponding basic compound planners design $\kappa_{cb,cons}/\kappa_{cb,aggr}$ in our framework (i.e., without information filter and aggressive unsafe set) and the ultimate compound planners design $\kappa_{cu,cons}/\kappa_{cu,aggr}$ in our framework (i.e., with all techniques), under various communication settings.

We design three types of communication. In the "no disturbance" setting, the ego vehicle $C_0$ always obtains messages from $C_1$ every $\Delta t_m$ seconds. In the "messages delayed" setting, messages may delay with $\Delta t_d$ or drop with a probability $p_d$, where $\Delta t_d = 0.25s$ and $p_d = \{0.05j \mid j = 0, 1, \ldots, 19\}$. In the "messages lost" setting, messages always drop, and thus only sensor information is available. The sensor uncertainty is $\delta_p = \delta_v = \delta_a = \{1 + 0.2j \mid j = 0, 1, \ldots, 19\}$. For each type of communication, we conduct 80,000 simulations. In each simulation, we randomly generate a sequence of accelerations in which the $i$-th element is the control input of $C_1$ at the $i$-th timestamp. The initial positions are $p_0(0) = -30m$ and $p_1(0) = \{50.5 + 0.5j \mid j = 0, 1, \ldots, 19\}$, and the position of the unsafe area is between $5m$ and $15m$. We assume $\Delta t_c = 0.05s$ and $\Delta t_m = \Delta t_s$.

TABLE I: The comparison between the conservative pure NN-based planner $\kappa_{n,cons}$ and the corresponding basic and ultimate compound planners. Winning percentage is the percentage of simulations in which the ultimate compound planner has the higher $\eta$ value. Emergency frequency is the percentage of timestamps at which the acceleration is determined by $\kappa_e$.

| settings | planner types | reaching time | safe rate | $\eta$ value | winning percentage | emergency frequency |
|---|---|---|---|---|---|---|
| no disturbance | pure NN | 7.989s | 100% | 0.144 | 99.97% | — |
| | basic | 7.990s | 100% | 0.144 | 99.97% | 0.01% |
| | ultimate | **6.408s** | 100% | **0.178** | — | 7.99% |
| messages delayed | pure NN | 8.868s | 100% | 0.127 | 100% | — |
| | basic | 8.871s | 100% | 0.127 | 100% | 0.03% |
| | ultimate | **6.719s** | 100% | **0.171** | — | 10.37% |
| messages lost | pure NN | 9.704s | 100% | 0.113 | 100% | — |
| | basic | 9.707s | 100% | 0.113 | 100% | 0.02% |
| | ultimate | **7.654s** | 100% | **0.150** | — | 17.58% |

TABLE II: The comparison between the aggressive NN-based planner $\kappa_{n,aggr}$ and the corresponding compound planners. '*' indicates that only reaching time of safe cases is counted.

| settings | planner types | reaching time | safe rate | $\eta$ value | winning percentage | emergency frequency |
|---|---|---|---|---|---|---|
| no disturbance | pure NN | ***4.513s** | 61.50% | −0.244 | 93.52% | — |
| | basic | 6.325s | **100%** | 0.177 | 99.66% | 20.39% |
| | ultimate | 6.130s | **100%** | **0.183** | — | 19.08% |
| messages delayed | pure NN | ***4.684s** | 56.04% | −0.314 | 93.57% | — |
| | basic | 6.766s | **100%** | 0.167 | 99.85% | 23.62% |
| | ultimate | 6.431s | **100%** | **0.176** | — | 21.39% |
| messages lost | pure NN | ***5.238s** | 59.02% | −0.289 | 82.98% | — |
| | basic | 7.769s | **100%** | 0.145 | 99.87% | 29.67% |
| | ultimate | 7.385s | **100%** | **0.154** | — | 29.14% |

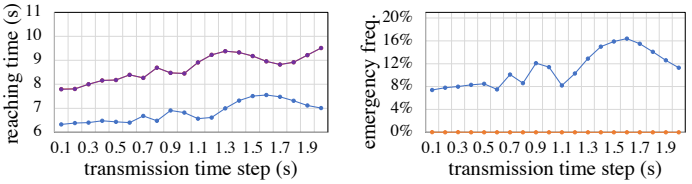
Table I shows the results of planners based on $\kappa_{n,cons}$. As the reaching time of $\kappa_{cb,cons}$ is almost the same as that of $\kappa_{n,cons}$, our basic compound planner design causes no efficiency degradation. Moreover, with the aid of the information filter and the aggressive unsafe set, the reaching time of the ultimate compound planner $\kappa_{cu,cons}$ is significantly reduced.

Table II shows the results of planners based on $\kappa_{n,aggr}$. Although the pure aggressive NN-based planner $\kappa_{n,aggr}$ has the shortest reaching time, a collision occurs in around 40% of the simulations. On the contrary, our compound planners achieve 100% safe rate. Besides, as we only count the reaching time of those safe simulations, the efficiency difference between our planner and the NN-based planner is actually smaller than it appears. The winning percentage also indicates our planner shows greater safety and efficiency than the NN-based planner (i.e., $\eta(\kappa_{n,aggr}) \le \eta(\kappa_{cu,aggr})$) in more than 80% of simulations.

From these two tables, we can see that **our framework provides significant improvements over the pure NN-based planners in safety and efficiency under various communication scenarios**.

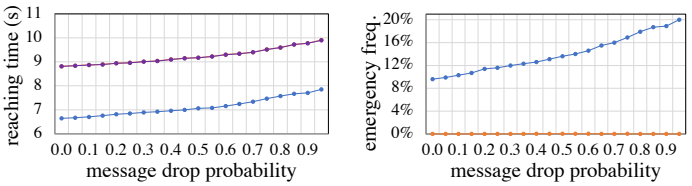
### B. Impact of Communication Disturbance

In Figures 5a, 5c, and 5e, we analyze the relation between reaching time and communication disturbance. As expected, if $C_0$ has received less information (i.e., larger transmission time steps or higher message drop probability) or the information becomes less precise (i.e., higher sensor uncertainty), the efficiency of $C_0$ decreases. Nevertheless, our ultimate compound planner still outperforms the pure NN-based planner by a significant margin when communication disturbance is severe. On the other hand, if the amount and quality of the received information deteriorate, the estimated unsafe set could expand. Consequently, it is more easily for $C_0$ to run into the emergency mode, as illustrated in Figures 5b, 5d, and 5f. Similarly, Tables I and II show that a



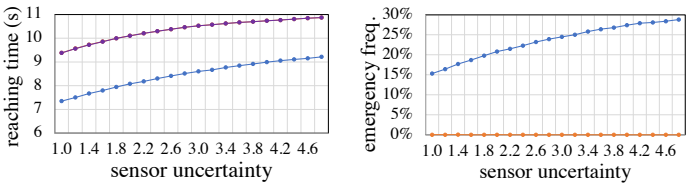

(a) Reaching time to transmission time step.

(b) Emergency frequency to transmission time step.




(c) Reaching time to message drop probability.

(d) Emergency frequency to message drop probability.

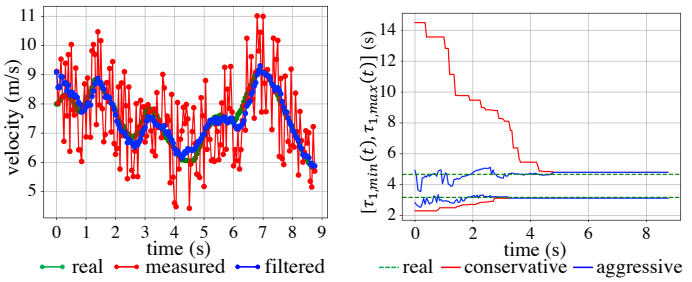



(e) Reaching time to sensor uncertainty.

(f) Emergency frequency to sensor uncertainty.

pure NN basic ultimate

Fig. 5: The impact of communication disturbance on the performance of $\kappa_{n,cons}$, $\kappa_{cb,cons}$, and $\kappa_{cu,cons}$.




(a) Measured velocities before and after filter.

(b) Conservative and aggressive passing time window estimation.

Fig. 6: Effectiveness of info. filter and aggressive unsafe set.

more aggressive NN-based planner could also lead to a higher emergency frequency.

### C. Effectiveness of Information Filter and Aggressive Unsafe Set

As described in Section III, the objective of the information filter and the aggressive unsafe set is to obtain a small and precise estimation of the unsafe set. Figure 6a demonstrates an example of sensor-measured velocities before and after the filter. As we can see, the filtered results are quite close to the real velocities of $C_1$, though the sensor-measured values are quite inaccurate. In addition, we sample 200 trajectories of $C_1$ and compute the root-mean-square-error (RMSE) before and after the filter. It turns out that the RMSE of $C_1$'s position (*resp.* velocity) reduces by 69% (*resp.* 76%) after leveraging the filter. Thus, the information filter increases the precision for the estimation of the unsafe set.

Figure 6b illustrates the conservative (as Equation (7)) and the aggressive (as Equation (8)) estimations of $C_1$'s passing time window. The aggressive time window is much more compact than the conservative one, and it is pretty close to the real passing time. The result indicates our aggressive unsafe set estimation method could effectively capture the core part of the unsafe set, thus increasing the system efficiency.

## VI. Conclusion

We propose a safety-guaranteed framework for NN-based planners in connected vehicle environments under communication disturbance. In our framework, the given NN-based planner is combined with a runtime monitor and an emergency planner to form a compound planner that guarantees system safety. With the aid of the information filter and the aggressive unsafe set, the system efficiency is further enhanced. In a case study of unprotected left turn, our framework always assures safety even under severe communication disturbance, while showing similar or better efficiency than NN-based planners in most of the time.

## Acknowledgment

This work is supported by Ministry of Education (MOE) in Taiwan under grant NTU-111V1901-5, National Science and Technology Council (NSTC) in Taiwan under grant NSTC-111-2636-E-002-018, and Qualcomm, United States National Science Foundation (NSF) grants 1834701 and 1724341, and United States Office of Naval Research (ONR) grant N00014-19-1-2496.

## References

[1] J. Dong, S. Chen, Y. Li, R. Du, A. Steinfeld, and S. Labi, "Facilitating connected autonomous vehicle operations using space-weighted information fusion and deep reinforcement learning based control," *arXiv preprint arXiv:2009.14665*, 2020.

[2] P. Y. J. Ha, S. Chen, J. Dong, R. Du, Y. Li, and S. Labi, "Leveraging the capabilities of connected and autonomous vehicles and multi-agent reinforcement learning to mitigate highway bottleneck congestion," *arXiv preprint arXiv:2010.05436*, 2020.

[3] M. Zhou, Y. Yu, and X. Qu, "Development of an efficient driving strategy for connected and automated vehicles at signalized intersections: A reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, vol. 21, no. 1, pp. 433–443, 2020.

[4] Z. Cao, E. Bıyık, W. Z. Wang, A. Raventos, A. Gaidon, G. Rosman, and D. Sadigh, "Reinforcement learning based control of imitative policies for near-accident driving," in *Proceedings of Robotics: Science and Systems (RSS)*, 2020, pp. 39:1–39:10.

[5] J. Wang, Y. Wang, D. Zhang, Y. Yang, and R. Xiong, "Learning hierarchical behavior and motion planning for autonomous driving," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2235–2242.

[6] X. Liu, C. Huang, Y. Wang, B. Zheng, and Q. Zhu, "Physics-aware safety-assured design of hierarchical neural network based planner," in *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*, 2022, pp. 137–146.

[7] X. Liu, R. Jiao, B. Zheng, D. Liang, and Q. Zhu, "Safety-driven interactive planning for neural network-based lane changing," *arXiv preprint arXiv:2201.09112*, 2022.

[8] C. Huang, S. Xu, Z. Wang, S. Lan, W. Li, and Q. Zhu, "Opportunistic intermittent control with safety guarantees for autonomous systems," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 37.5:1–37.5:6.

[9] C. Huang, J. Fan, X. Chen, W. Li, and Q. Zhu, "POLAR: A polynomial arithmetic framework for verifying neural-network controlled systems," in *International Symposium on Automated Technology for Verification and Analysis*, 2022, pp. 414–430.

[10] S. Oh, L. Zhang, E. Tseng, W. Williams, H. Kourous, and G. Orosz, "Safe decision and control of connected automated vehicles for an unprotected left turn," in *Dynamic Systems and Control Conference*, 2020, V001T10A005.

[11] T. Awal, M. Murshed, and M. Ali, "An efficient cooperative lane-changing algorithm for sensor- and communication-enabled automated vehicles," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 1328–1333.

[12] B. Zheng, C.-W. Lin, S. Shiraishi, and Q. Zhu, "Design and analysis of delay-tolerant intelligent intersection management," *ACM Transactions on Cyber-Physical Systems (TCPS)*, vol. 4, no. 1, pp. 1–27, 2019.

[13] M. Xu, Y. Luo, G. Yang, W. Kong, and K. Li, "Dynamic cooperative automated lane-change maneuver based on minimum safety spacing model," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 1537–1544.

[14] S. D. Bopardikar, B. Englot, and A. Speranzon, "Robust belief roadmap: Planning under uncertain and intermittent sensing," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6122–6129.

[15] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995.

[16] A. Becker (www.kalmanfilter.net), "Online kalman filter tutorial," accessed: 2022-09-24. [Online]. Available: https://www.kalmanfilter.net/