USENIX Security '25 Artifact Appendix - From Threat to Trust: Exploiting Attention Mechanisms for Attacks and Defenses in Cooperative Perception

Chenyi Wang[1] Raymond Muller[2] Ruoyu Song[2] Jean-Philippe Monteuuis[3]
Jonathan Petit[3] Yanmao Man[5] Ryan Gerdes[4] Z. Berkay Celik[2] Ming Li[1]
[1]University of Arizona [2]Purdue University [3]Qualcomm [4]Virginia Tech [5]Independent Researcher, U.S.
{chenyiw, yman, lim}@arizona.edu, {mullerr, song464, zcelik}@purdue.edu
{jmonteuu, petit}@qti.qualcomm.com, rgerdes@vt.edu

A Artifact Appendix

A.1 Abstract

This artifact provides the implementation and evaluation framework for the paper accepted at USENIX Security 2025 Cycle 2 - From Threat to Trust: Exploiting Attention Mechanisms for Attacks and Defenses in Cooperative Perception, which introduces SOMBRA, a novel cooperative perception attack, and LUCIA, a novel light-weight defense. The artifact includes code for generating adversarial perturbations, using both SOMBRA and the prior art method, evaluating their impact on state-of-the-art cooperative perception models. The codebase also includes the implementation for the defense LUCIA, compared with the official ROBOSAC implementation. The artifact also includes a case study on traffic jam scenarios. The provided instructions enable evaluators to reproduce the results and validate the claims made in the paper.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

There are no known security, privacy, or ethical concerns associated with running this artifact. The artifact does not require disabling security mechanisms or accessing sensitive data. All datasets used are publicly available and properly cited.

A.2.2 How to access

The artifact is hosted on GitHub and can be accessed at https://github.com/WiSeR-Lab/SOMBRA_LUCIA. *A stable version of the artifact, including instructions and the collected custom dataset, is archived on Zenodo at* https://doi.org/10.5281/zenodo.15523768 *and* https://zenodo.org/records/15523769, *respectively.*

A.2.3 Hardware dependencies

The artifact requires a machine with the following specifications:

- GPU: NVIDIA GPU with CUDA support (tested on CUDA 11.8 and 12.0).
- Minimum 8 GB of GPU memory.
- Minimum 16 GB of system memory.
- Disk space: At least 50 GB for datasets, models, and intermediate results.

A.2.4 Software dependencies

The artifact requires the following software:

- Operating System: Linux (tested on Ubuntu 20.04 and 22.04).
- Python: Version 3.8 or higher.
- Package manager: `pixi` for environment setup (https://pixi.sh).
- PyTorch: Version compatible with the CUDA version (e.g., PyTorch with CUDA 11.8 or 12.0).
- Additional Python packages: Listed in the `pixi.toml` file included in the repository.

A.2.5 Benchmarks

The artifact uses the following datasets and models:

- Dataset: OPV2V test split, available at https://mobility-lab.seas.ucla.edu/opv2v/.
- Pretrained models: Attentive Fusion, CoAlign, Where2comm, and V2VAM, available at https://github.com/DerrickXuNu/OpenCOOD/tree/main.
- Traffic jam case study dataset: Available at https://zenodo.org/records/15523769. A data and model downloading script is provided with the artifact, with instructions given in A.3.3.

A.3 Set-up

A.3.1 Installation

Follow these steps to set up the environment:
1. Install `pixi` by following the instructions at https://pixi.sh.
2. Clone the artifact repository:

```
git clone \
https://github.com/WiSeR-Lab/SOMBRA_LUCIA.git
cd SOMBRA_LUCIA
```

3. Activate the environment and install dependencies:

```
# One-liner setting up and activating VE
pixi shell
# Setup opencood package
python setup.py develop
# Build IoU operation on CUDA
python opencood/utils/setup.py build_ext --inplace
```

A.3.2 Basic Test

Run the following command to verify the setup:

```
python cp_attack.py --help
```

Expected output: A list of available arguments and their descriptions. This confirms that the environment is correctly set up.

A.3.3 Dataset and Model Weights

We use the test split of the OPV2V dataset, and the official pretrained weights for Attentive Fusion, CoAlign, Where2comm, and V2VAM. The latest download links are available at https://github.com/DerrickXuNu/OpenCOOD.

We host the same dataset and model weights needed for the evaluation of SOMBRA and LUCIA, the latest link can be found at our GitHub repo: https://github.com/WiSeR-Lab/SOMBRA_LUCIA:

You can also use:

```
./scripts/data_model_download.sh
```

to download and extract the data and model weights in to `./assets/`.

The dataset after decompression has the following structure:

```
test
|-- 2021_08_18_19_48_05
|   |---- data_protocol.yaml
|   |---- 1045
|   |---- 1054
|...
|-- scenario_timestamp_id
|   |---- data_protocol.yaml
|   |---- agent_id_1
|   |---- ...
|   |---- agent_id_n
```

Note: *use the data root directory to specify dataset path (i.e., `--data_dir /some_path/test/`). When `--data_dir <path>` is passed to `cp_attack.py`, it overwrites the `validation_dir` specified in the model yaml config.*

The model weights after decompression has the following structure:

```
opv2v_weights
|-- attfusion
|   |---- latest.pth
|   |---- config.yaml
|-- coalign
|   |---- net_epoch15.pth
|   |---- config.yaml
|-- v2vam
|   |---- latest.pth
|   |---- config.yaml
|-- where2comm
|   |---- net_epoch50.pth
|   |---- config.yaml
```

Note: *use the model root directory to specify model path (i.e., `--model_dir /some_path/model_name/`). Also, the model name argument (i.e., `--model <AttentiveFusion/CoAlign/V2VAM/Where2comm>`) is case-sensitive and is used to build the correct attention-hijacking loss for SOMBRA.*

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): *SOMBRA achieves high success rates in targeted object removal (TOR) and mass object removal (MOR) attacks against state-of-the-art cooperative perception models. This is demonstrated by experiments (E1) and (E2) in Section 6.2 of the paper.*

(C2): *SOMBRA outperforms prior art in terms of attack success metrics. This is also demonstrated by experiments (E1) and (E2) in Section 6.2 of the paper.*

(C3): *SOMBRA remains effective in the traffic jam scenario with varying number of CAVs. This is demontrated by experiments (E3) in Section 6.3 of the paper.*

(C4): *LUCIA mitigates the impact of SOMBRA by restoring the detection and perception quality measured by mAP. This is demonstrated by experiments (E4) in Section 6.4 of the paper.*

A.4.2 Automated Evaluation Scripts

- *We aggregated dataset and model downloading process, and evaluation commands for different models into scripts located in* `./scripts/`
- *Use* `data_model_download.sh` *to download and process the dataset and weights.*
- *Use* `e1_e2.sh` *to obtain results for C1 and C2. Use* `e3.sh` *and* `e4.sh` *for C3 and C4, respectively.*
- *The results are stored under the same directory as the model weights, in* `./assets/opv2v_weights/<model>`.

A.4.3 Experiments

(E1): Targeted / Mass Object Removal [1 human-hour + 4 compute-hours]:
Preparation: *Download the OPV2V test split and pretrained models. Set the* `-data_dir` *and* `-model_dir` *arguments accordingly.*
Execution: *Run the following command:*

```
python cp_attack.py \
--model_dir <path_to_model> \
--model <cp_model_name> \
--data_dir <path_to_opv2v_test> \
--attack_mode <mor/tor> \
--loss sombra \
(--target_id <random/in/out>)
```

Alternatively, you can use `./scripts/e1_e2.sh` *to obtain the results.*
Results: *Check the saved results in the model directory. Verify the attack success rate matches the reported values in Table 1, 2 of the paper.*

(E2): Comparison with Prior Art [1 human-hour + 4 compute-hours]:
Preparation: *Same as (E1).*
Execution: *Run the following commands for prior art:*

```
python cp_attack.py \
--model_dir <path_to_model> \
--model <cp_model_name> \
--data_dir <path_to_opv2v_test> \
--attack_mode <mor/tor> \
--loss pa \
(--target_id <random/in/out>)
```

Alternatively, you can use `./scripts/e1_e2.sh` *to obtain the results.*
Results: *Compare the attack success rate metrics with SOMBRA, as shown in Table 1, 2 of the paper.*

(E3): Traffic Jam Case Study [2 human-hours + 4 compute-hours]:
Preparation: *Download the traffic jam dataset. Generate perturbations:*

```
python cp_attack.py \
--model_dir <path_to_model> \
--model <cp_model_name> \
--data_dir <path_to_traffic_jam> \
--attack_mode mor \
--loss sombra \
--save_perturb
```

Rename the folder storing perturbed features to `adv_feature`*. Also, verify that the attack success rates match the reported values in Table 3 of the paper.* Alternatively, you can use `./scripts/e3.sh` *to obtain the results.*
Execution: *Run the case study for analyzing adversarial signal propagation:*

```
python case_study.py \
--model_dir <path_to_attfusion> \
--data_dir <path_to_traffic_jam>
```

Results: *Verify the victim's attention paid to the attacker matches the pattern in Figure 9 of the paper, which is found in* `attn` *column of the result csv* `diff_cav.csv`

(E4): Defense Evaluation [2 human-hour + 8 compute-hours]:
Preparation: *Same as (E1).*
Execution: *Run the following command with defenses enabled:*

```
python cp_attack.py \
--model_dir <path_to_model> \
--model <cp_model_name> \
--data_dir <path_to_opv2v_test> \
--attack_mode <mor/tor> \
--loss <sombra/pa> \
(--target_id <random/in/out>) \
(--defense or --robosac)
```

Alternatively, you can use `./scripts/e4.sh` *to obtain the results.*
Results: *Verify the reduced attack success rates with LUCIA and ROBOSAC, as reported in Table 4, 5 of the paper.*

A.5 Notes on Reusability

The artifact is modular and can be extended to evaluate other cooperative perception models or attack strategies. The `cp_attack.py` script can be adapted to test new loss functions or defenses. The traffic jam case study framework can be reused for other scenarios involving cooperative perception.

A.6 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2025/.