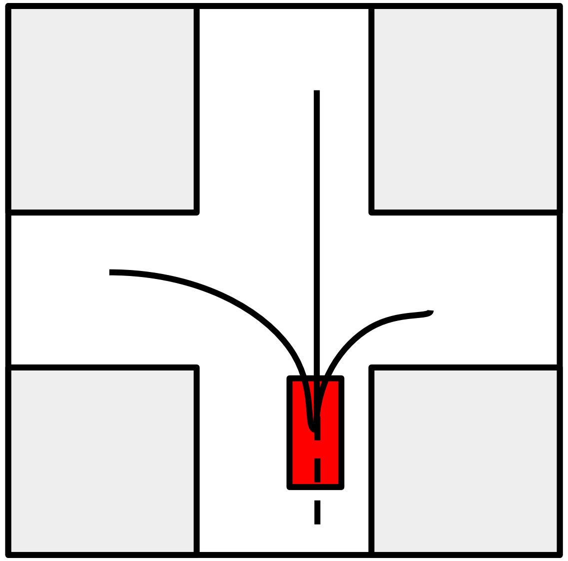
Ellipse Loss for Scene-Compliant Motion Prediction

Henggang Cui*, Hoda Shajari*, Sai Yalamanchi, Nemanja Djuric

Trajectory prediction for autonomous driving

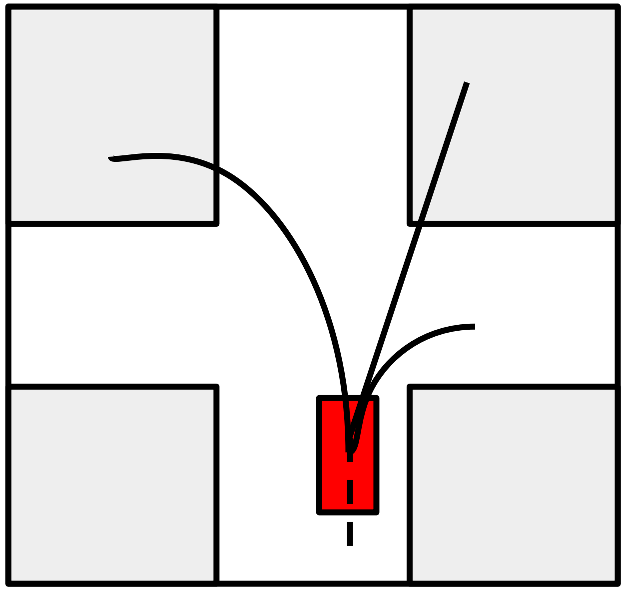
- Predict the trajectories of the other traffic actors in the scene
  - Critical task for autonomous driving
  - State-of-the-art is to use a deep neural network
  - Takes history observations and map as the input
  - Outputs the future trajectory predictions as sequences of waypoints (x, y, θ)
  - Trained with distance-based loss (e.g., $L2(x_{pred}, x_{gt}) + L2(y_{pred}, y_{gt})$)



UBER ATG

Scene-compliant trajectory prediction

- The trajectories should be compliant to road geometries (i.e., *scene-compliant*)
    - E.g., a vehicle is unlikely to go outside the road surface
    - Traditional losses that are based on distance between prediction and ground-truth don’t explicitly penalize off-road predictions
- We propose *ellipse loss* that explicitly penalizes off-road predictions

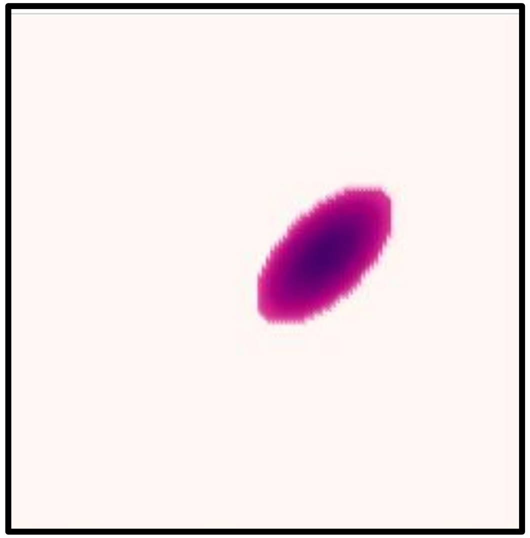


Non-scene-compliant predictions that go off-road aren’t penalized enough by traditional losses

Ellipse loss

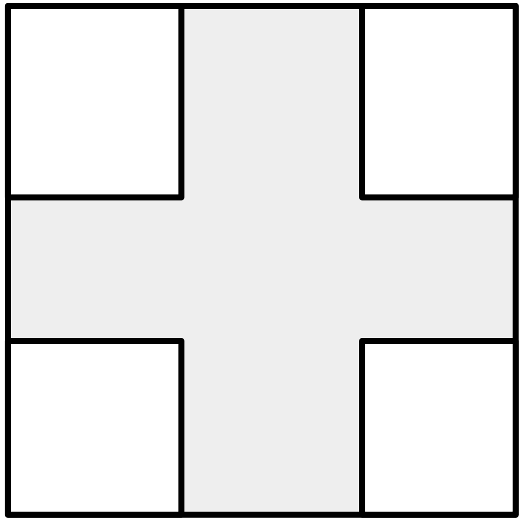
- Give a waypoint prediction (x, y, θ) and a non-drivable area mask D
- "Rasterize" waypoint into an occupancy grid G in a differentiable manner
- Compute the ellipse loss as $G * \mathrm{D}$

$$\mathcal{L}_{\text{ellipse}}(x, y, \theta, D) =$$



occupancy grid (G)

x



non-drivable area (D)

Differentiable rasterization

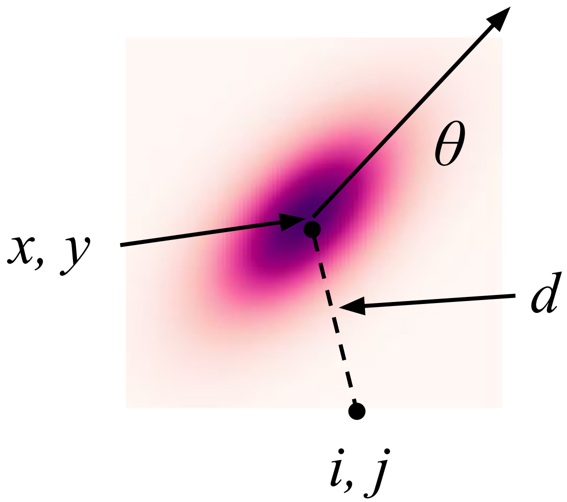
- Rasterize a waypoint (x, y, θ) to an occupancy grid G
- Compute the value of each cell G_{ij} as the density of a 2D Gaussian $N(0, \Sigma)$ evaluated at $d_{ij}(x, y)$
  - where $d_{ij}(x, y)$ is the distance vector from cell (i, j) to the waypoint (x, y, θ)

$$\mathcal{G}_{i,j}(x,y,l,w,\theta) = \mathcal{N}\left(\mathbf{d}_{i,j}(x,y); \mathbf{0}, \mathbf{\Sigma}(l,w,\theta)\right)$$

- Σ depends on the size and orientation θ of the actor bounding box
  - Rotates the Gaussian ellipse to the box orientation

$$\mathbf{\Sigma}(l,w,\theta) = \mathbf{R}(\theta)^{\top} \mathrm{diag}(\sigma_l, \sigma_w)\mathbf{R}(\theta)$$

- This allows us to compute the gradients of G w.r.t. (x, y, θ)



UBER ATG

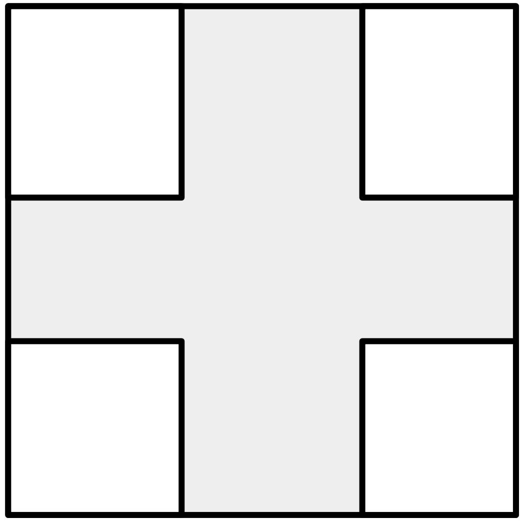
Compute ellipse loss

- Compute the loss as $G * D$
  - The loss will the Gaussian probability mass away from the non-drivable area by shifting and rotating the waypoint

$$\mathscr{L}_{\text{ellipse}}(x, y, \theta, D) =$$

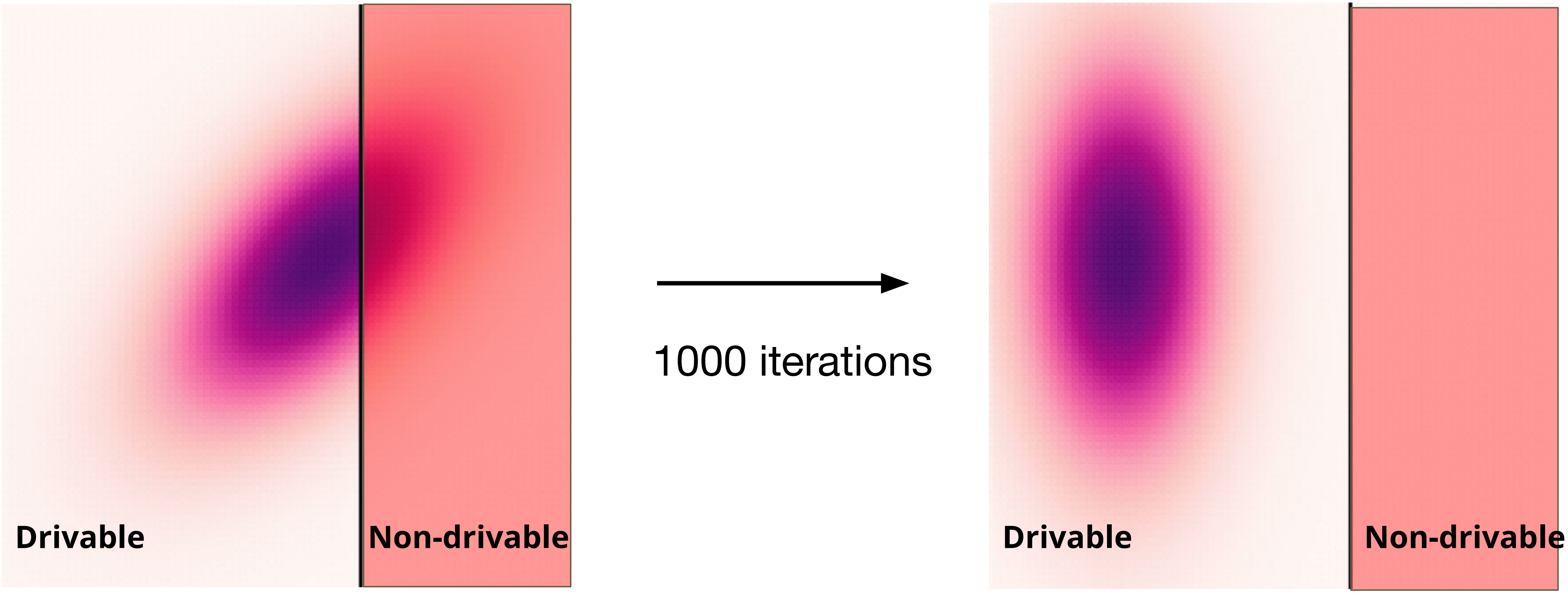
occupancy grid (G)

x



non-drivable area (D)

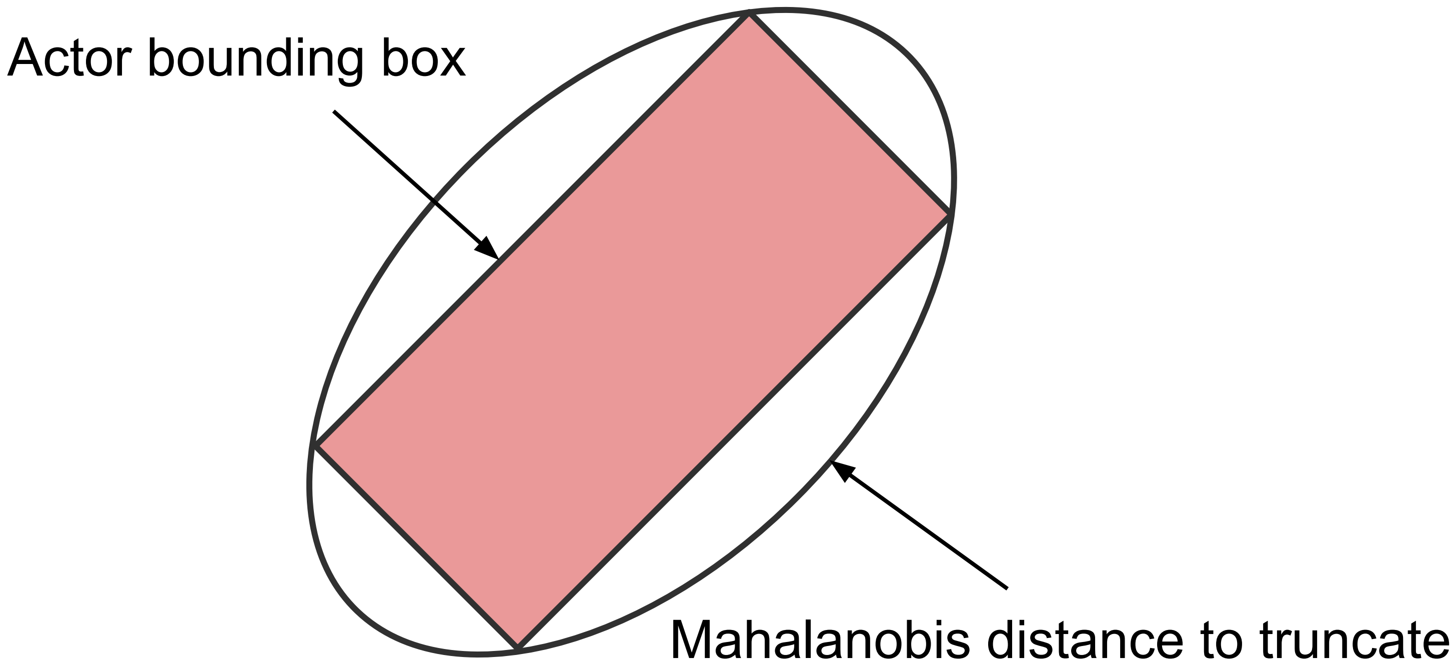
Toy example

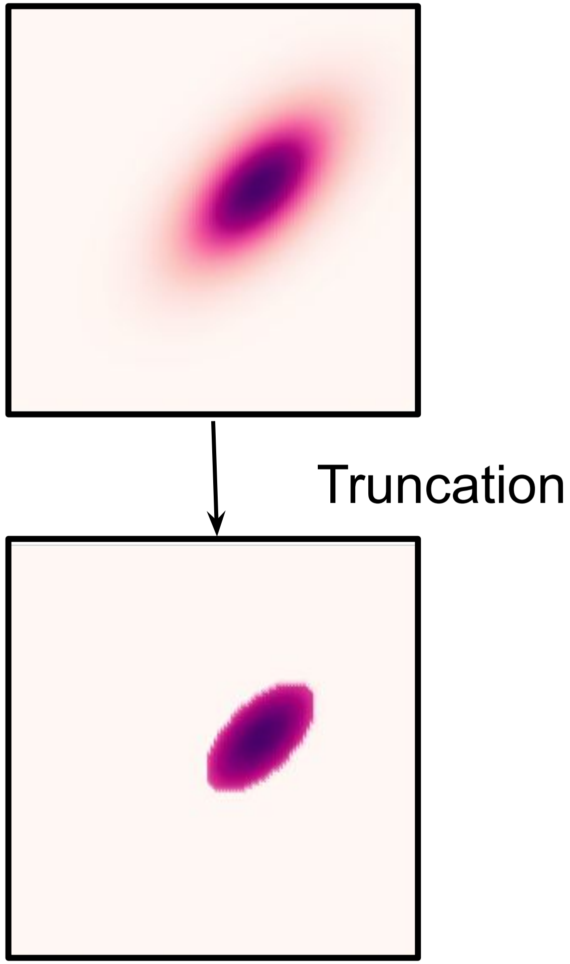


- Update the waypoint parameter (x, y, θ) using ellipse loss for 1000 iterations
- Waypoint leaves non-drivable area with both translation and rotation

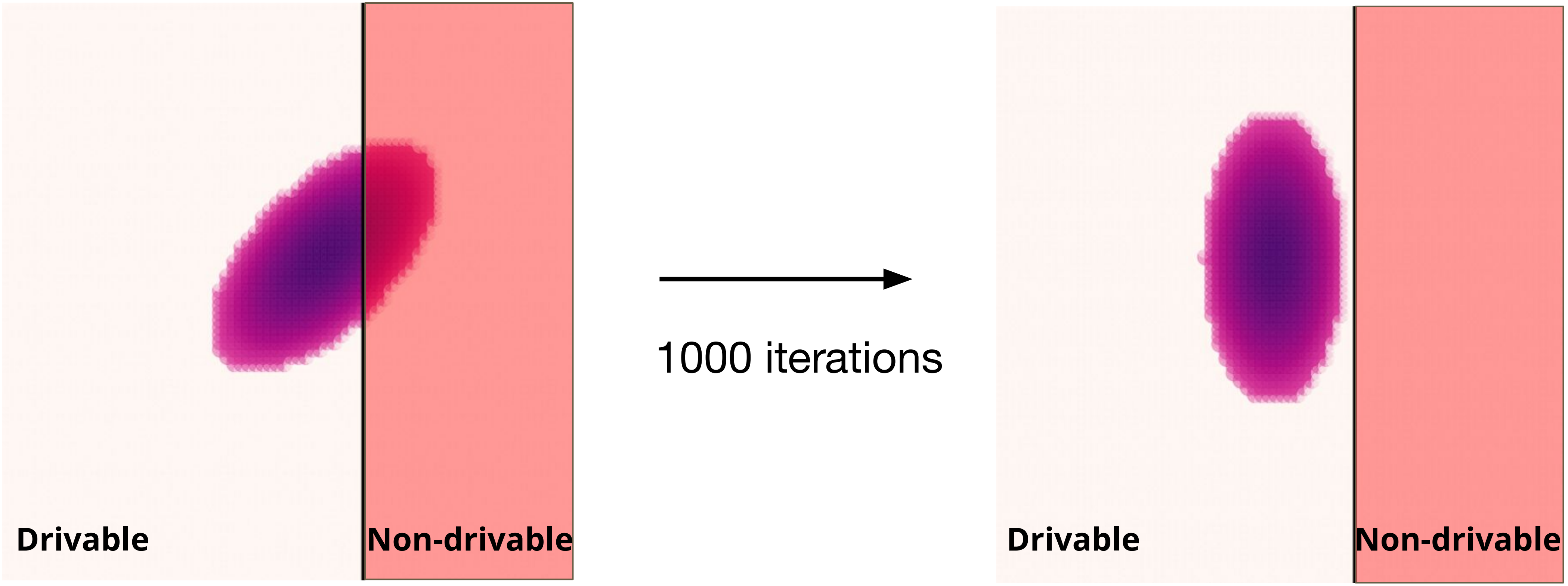
Truncate G at box boundary

- Truncate G at a Mahalanobis distance that exactly covers the bounding box
- Values outside the range will not contribute to the loss
- So that ellipse loss only penalizes off-road predictions





Toy example of ellipse loss with the truncation



- Waypoint leaves non-drivable area with both translation and rotation
- It stops as soon as it leaves the drivable area instead of keeping moving away

UBER ATG

Implementation details and baselines

- Implemented ellipse loss in MultiXNet [1]
    - Jointly predict the trajectories of all actors in the scene
    - Predict 3 second future trajectories using 1 second history
    - Added ellipse loss as an additional loss term in addition to the standard distance-based prediction loss
- Baselines
    - Vanilla MultiXNet with only the vanilla prediction loss
    - Off-road upweighting baseline [2]
        - Upweights the vanilla prediction loss by 5.0 for off-road predictions

[1] Djuric et al. MultiXNet: Multiclass Multistage Multimodal Motion Prediction.

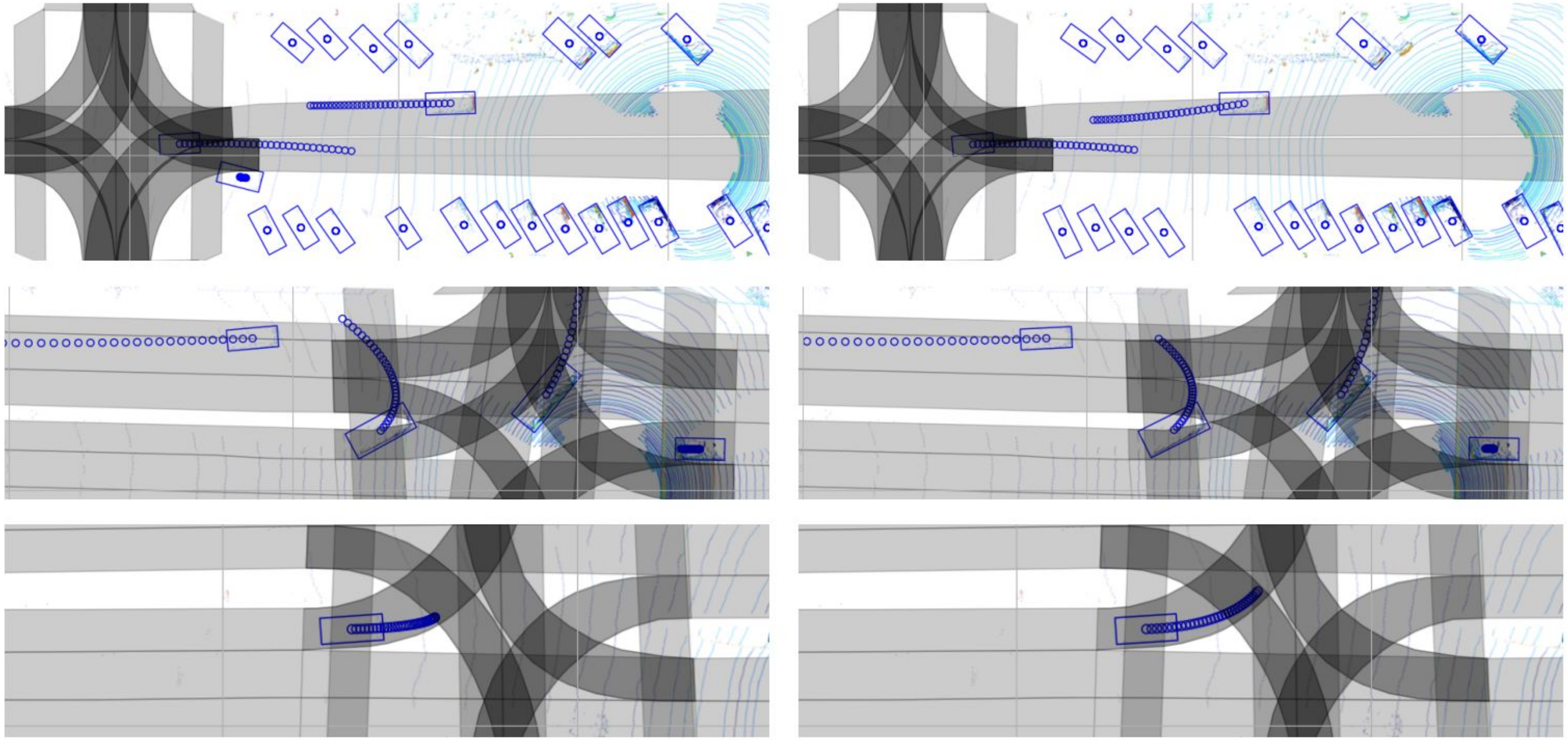
[2] Niedoba et al. Improving Movement Prediction of Traffic Actors using Off-road Loss and Bias Mitigation

Quantitative results

Method	ℓ_2 [m] ↓ Avg	ℓ_2 [m] ↓ @3s	CtrORFP [%] ↓ Avg	CtrORFP [%] ↓ @3s	BoxORFP [%] ↓ Avg	BoxORFP [%] ↓ @3s
MultiXNet [6]	**0.465** ± **0.001**	**0.836** ± **0.002**	0.061 ± 0.007	0.094 ± 0.018	0.624 ± 0.009	0.850 ± 0.029
Off-Road [25]	**0.465** ± **0.002**	**0.837** ± **0.005**	0.058 ± 0.001	0.086 ± 0.006	0.616 ± 0.015	0.853 ± 0.020
Ellipse-Loss	0.472 ± 0.004	0.846 ± 0.008	**0.031** ± **0.003**	**0.043** ± **0.006**	**0.445** ± **0.007**	**0.485** ± **0.004**

- Off-road (ORFP) metrics
  - (# of off-road predictions) / (# of all predictions)
  - CtrORFP: a waypoint is off-road if its center is off-road
  - BoxRFP: a waypoint is off-road if any of its four corners is off-road
- Ellipse loss reduces off-road predictions by about 50%

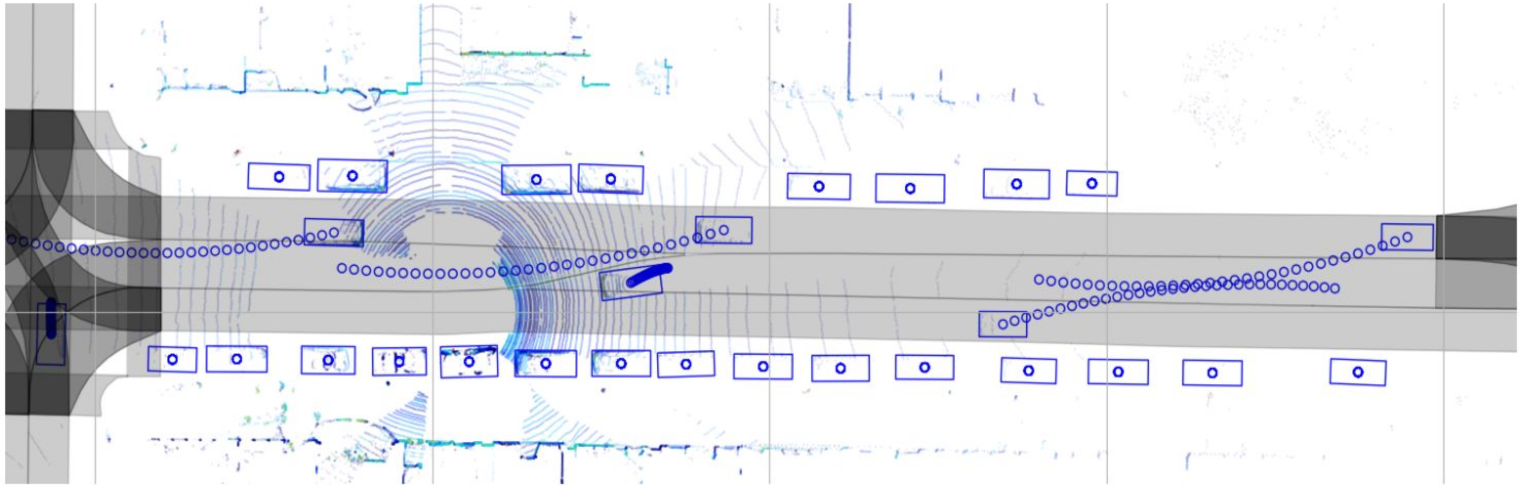
Qualitative results



Baseline **Ellipse loss**

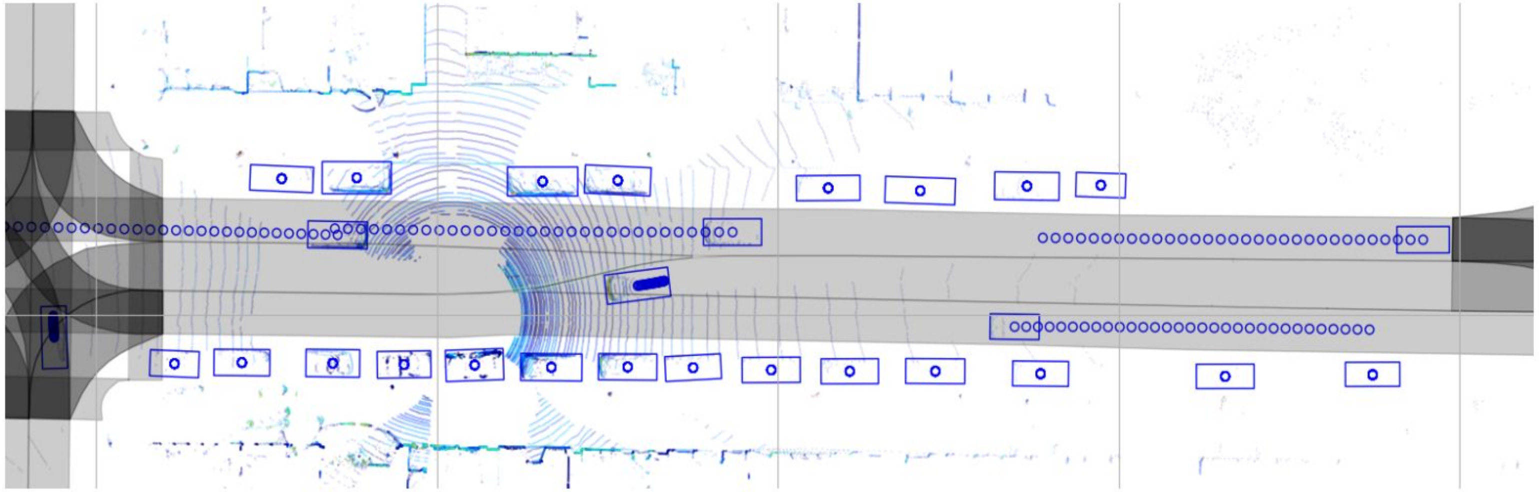
- Predicted trajectories are more scene-compliant with ellipse loss

Ablation studies



Ellipse loss without truncation

- Predictions go to the center of the road, causing high L2 error



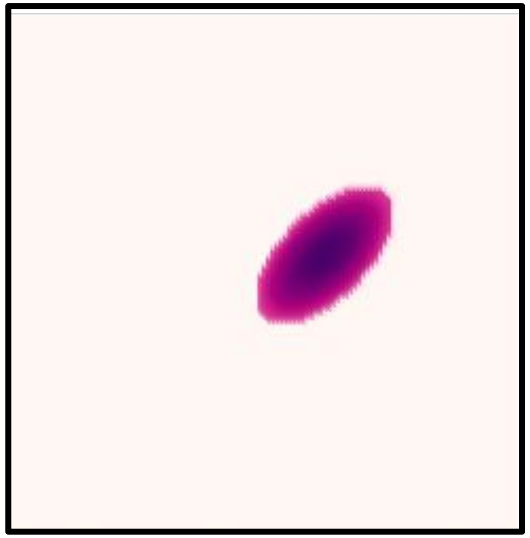
Ellipse loss with truncation

- Ellipse loss does not penalize predictions in drivable area

Conclusions

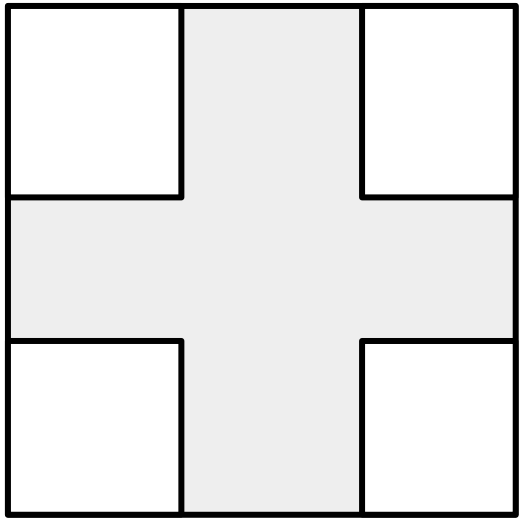
- Ellipse loss for scene-compliant motion trajectory prediction
- Reduces off-road predictions by 50%

$$\mathcal{L}_{\text{ellipse}}(x, y, \theta, D) =$$



occupancy grid (G)

x



non-drivable area (D)