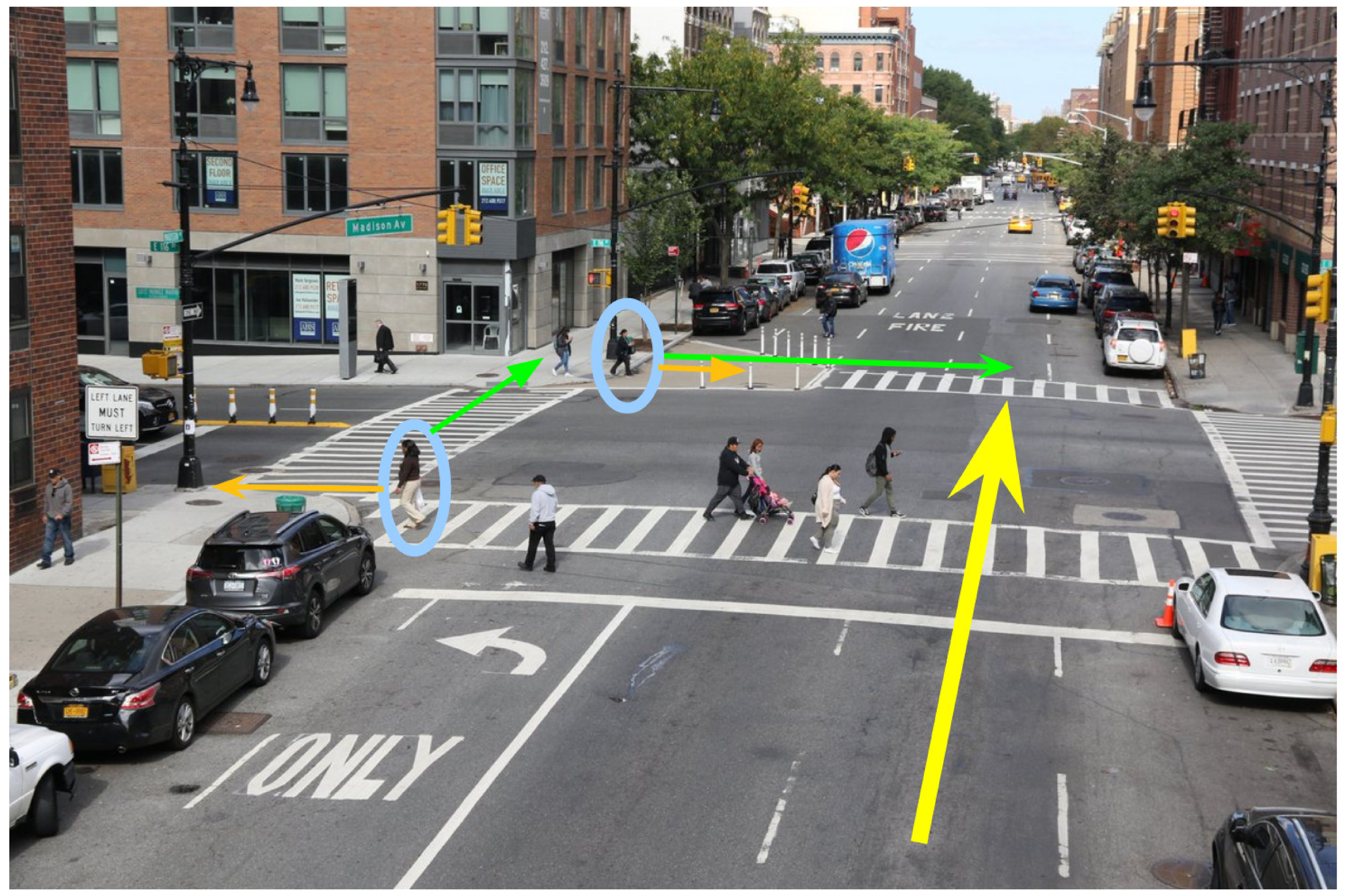
Control-Aware Prediction Objectives for Autonomous Driving

Rowan McAllister [1] Blake Wulfe [1] Jean Mercat [1] Logan Ellis [1] Sergey Levine [2] Adrien Gaidon [1]

[1]Toyota Research Institute [2]University of California, Berkeley

Motivation



- Consider an autonomous vehicle planning to drive along the yellow arrow.
- It forecasts each pedestrian's trajectory, with errors between prediction and ground truth
- **Question**: Which forecasting errors matter most here (have real-life consequences)?

- Problem: forecasting metrics typically unaware of usage ("objective mismatch" [9])
- Solution: weight forecasting metrics by their effect on downstream control
- Benefit: improves forecasting accuracy where it matters most (e.g. potential collisions)

The Literature

Control-Unaware Prediction Objectives

Common prediction metrics in the literature and in prediction benchmarking challenges–including Argoverse Forecasting [3], Lyft Prediction [7], and Waymo Open Motion [5]–are:

Metric name	Objective
Average Displacement Error (ADE)	$\|\|\hat{\mathbf{y}}_{1:T} - \mathbf{y}_{1:T}\|\|_2$
Final Displacement Error (FDE)	$\|\|\hat{\mathbf{y}}_{T} - \mathbf{y}_{T}\|\|_2$
Minimum-ADE (minFDE)	$\min_{k\in\{1,...,K\}} \|\|\hat{\mathbf{y}}^{(k)}_{1:T} - \mathbf{y}_{1:T}\|\|_2$
Minimum-FDE (minFDE)	$\min_{k\in\{1,...,K\}} \|\|\hat{\mathbf{y}}^{(k)}_{T} - \mathbf{y}_{T}\|\|_2$
Miss Rate (MR)	$\frac{1}{K}\sum_k \mathbb{1}[\alpha < \|\|\hat{\mathbf{y}}^{(k)}_{T} - \mathbf{y}_{T}\|\|_2]$
Negative Log Likelihood (NLL)	$-\log q(\mathbf{y}_{1:T})$

Control-Aware Prediction Objectives

Some common assumptions when solving the objective mismatch problem:

1. Is the planner **differentiable**? (useful for end2end methods and sensitivity analysis [6, 2])
2. Is the planner **stochastic**? (useful for policy gradient methods [8, 1])
3. Is the planner a **known** function? (useful for computing counterfactual actions)

We assume (3) only, since many real autonomous vehicle planners are human-designed for reasons of safety and verification. So our method can handle planners that are differentiable, non-differentiable, stochastic, or deterministic.

Our Method: Attention CAPO

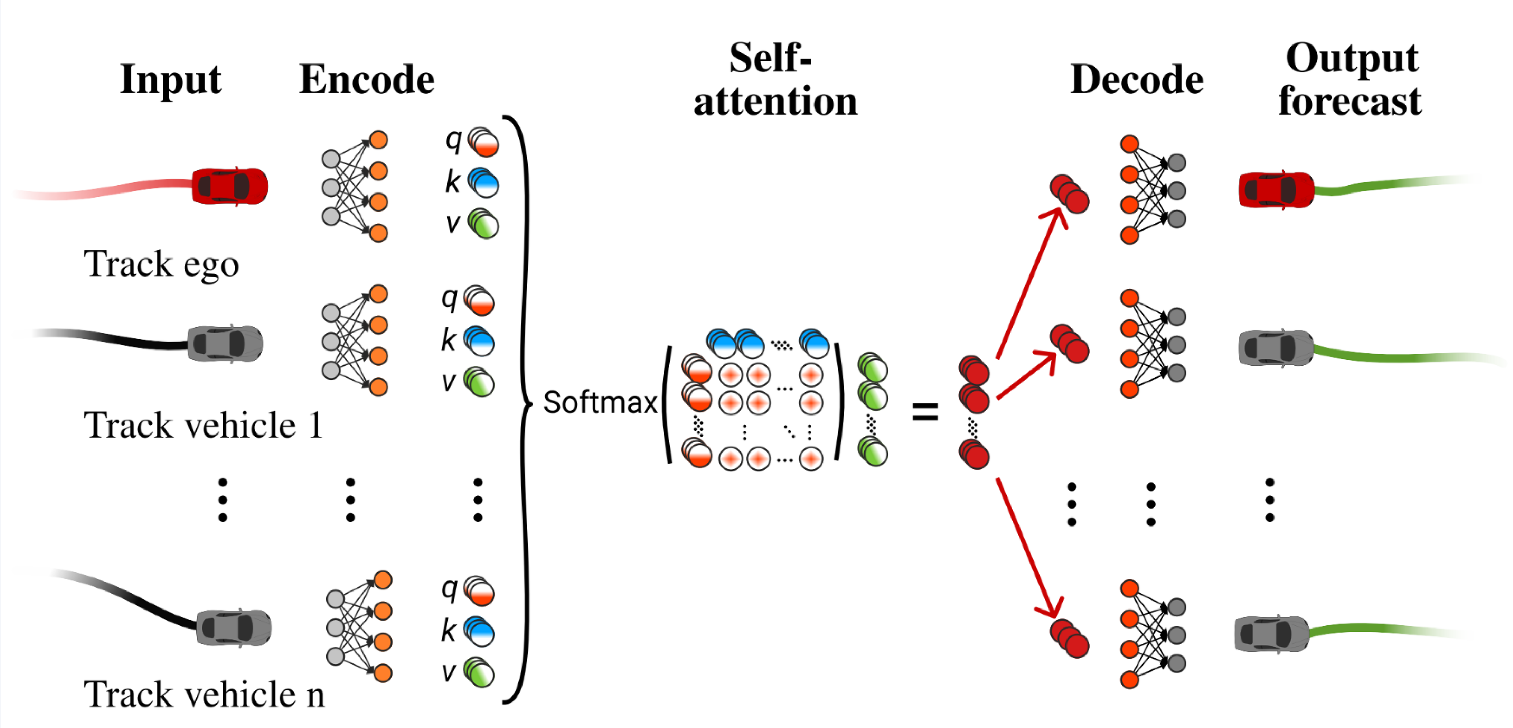


Figure 1. The equivariant attention weighting method uses the attention matrix from multi-agent trajectory forecasting, which reflects how much the ego vehicle's trajectory is a function of the other vehicles or pedestrians surrounding it.

$$\alpha(\mathbf{x}) = \sigma\left(\frac{Q(\mathbf{x})K(\mathbf{x})^\top}{\sqrt{d_k}}\right) = [\alpha_0, ..., \alpha_N], \quad (1)$$

$$\hat{\mathbf{y}} = \alpha(\mathbf{x})V, \quad (2)$$

$$q_\theta : \mathcal{X} \to \mathcal{P}_{\mathcal{Y}_{\text{agent}} \times \mathcal{Y}_{\text{ego}}}, \quad (3)$$

$$\theta_{\text{ego}} \leftarrow \theta_{\text{ego}} + \nabla_{\theta_{\text{ego}}} \log q_\theta(\mathbf{y}_{\text{ego}}|\mathbf{x}), \quad (4)$$

$$\theta_{\text{agent}} \leftarrow \theta_{\text{agent}} + \alpha(\mathbf{x})\nabla_{\theta_{\text{agent}}} \log q_\theta(\mathbf{y}_{\text{agent}}|\mathbf{x}). \quad (5)$$

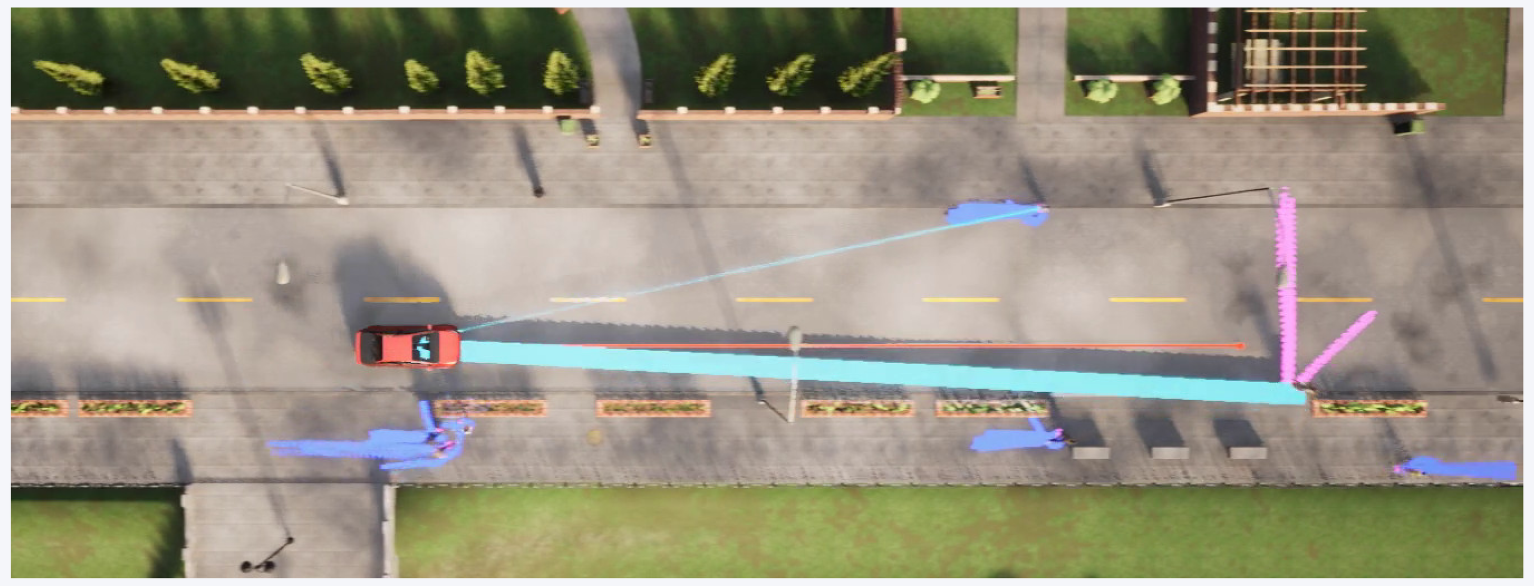


Figure 2. A vehicle drives to the right while reacting to pedestrians with sample predicted trajectories shown in blue or pink. Our Control-Aware Prediction Objectives (CAPO) can learn to capture which predictions should have more influence on the vehicle's controls (cyan line width proportional to attention weight).

Experiments

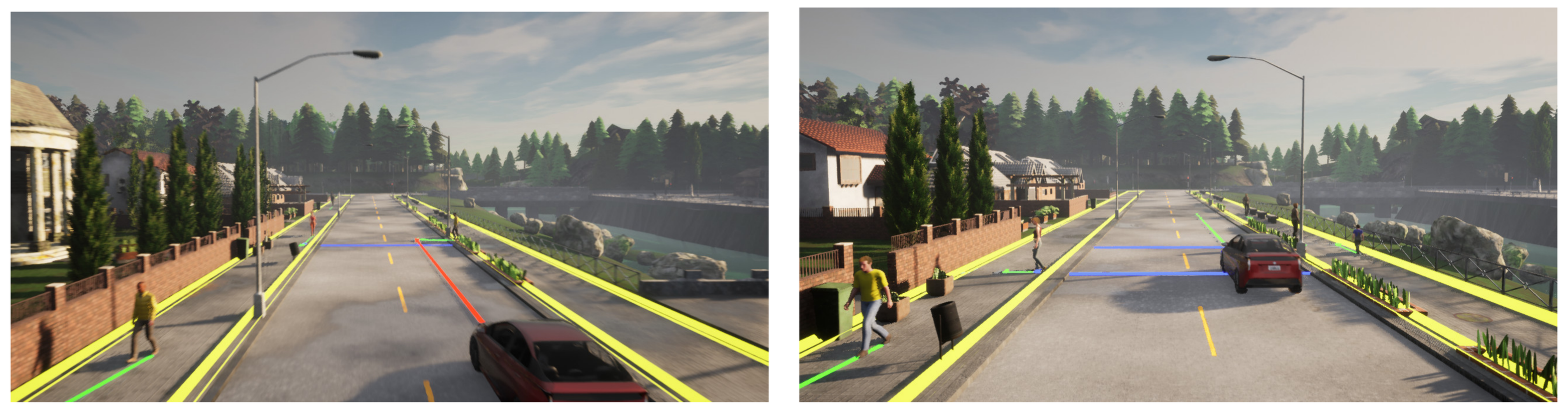


Figure 3. **Pedestrian Prediction Scenario.** We use the CARLA driving simulator [4]. Pedestrians spawn on the sidewalk (yellow region) and the ego (red) car predicts the pedestrian trajectories within the next 3 seconds (green). Some pedestrians will cross the road at right angles. **Left**: the planner predicts a collision with a crossing pedestrian and starts slowing (red ego drives up to the blue crossing line but not further). **Right**: ego is safely passing the road segment where the pedestrian has already crossed.

Our Method: Counterfactual CAPO

We can also weight errors by counterfactual action discrepancy. We isolate each pedestrian's *individual* contributions to the ego's control by combining how agent n *might* move $\hat{\mathbf{y}}^k_n \sim q_\theta(\mathbf{Y}_n|\mathbf{x})$ with how other agents *did* move:

$$\hat{\mathbf{u}}^k_n = \pi(\{\hat{\mathbf{y}}^k_n\} \cup \mathbf{y} \setminus \{\mathbf{y}_n\}), \quad (6)$$

and compare against the control had no agent deviated from their recorded trajectories:

$$\mathbf{u} = \pi(\mathbf{y}). \quad (7)$$

The difference corresponds to how much an individual agent affects the ego. For probabilistic models, multiple samples can ensure high importance even if agents only *might* affect control:

$$w_n = \max_{k\in\{1..K\}} \|\|\mathbf{u} - \hat{\mathbf{u}}^k_n\|\|_1, \quad (8)$$

which we use as weights for predictive model training:

$$\theta^* = \arg\max_\theta \sum_{n=1}^{N} w_n \log q_\theta(\mathbf{y}_n|\mathbf{x}). \quad (9)$$

Input: Controller: $\pi : \mathcal{X} \to \mathcal{U}$
1: Record trajectory data $\mathcal{D} = \{\mathbf{x}, \mathbf{y}\}_i$
2: **while** training **do**
3: Sample batch $\mathbf{x}, \mathbf{y} \sim \mathcal{D}$
4: Compute counterfactual controls: $\mathbf{u}, \hat{\mathbf{u}}^k_n$ ▷ Eq. (6)–(7)
5: Compute weight: $w(\mathbf{u}, \hat{\mathbf{u}}^k_n)$ ▷ Eq. (8)
6: Update model: $\theta \leftarrow \theta + w(\mathbf{u}, \hat{\mathbf{u}}^k_n)\nabla_\theta \log q_\theta(\mathbf{y}|\mathbf{x})$
Output: Predictive model $q_\theta : \mathcal{X} \to \mathcal{P}_{\mathcal{Y}}$

Results

Model	Objective	Collisions ↓	Speed (m/s) ↑	Jerk (m/s^{-3}) ↓	ADE (m) ↓	Control Error ↓
Baselines						
R2P2 [11]	$\ln q_\theta(\mathbf{y}\|\mathbf{x})$	11/100	9.97 ±0.222	8.92 ±0.250	**2.09** ±0.024	0.59 ±0.012
Attention [10]	$\ln q_\theta(\mathbf{y}_{\text{agent}}\|\mathbf{x}) + \ln q_\theta(\mathbf{y}_{\text{ego}}\|\mathbf{x})$	11/100	13.79 ±0.214	4.48 ±0.147	2.61 ±0.050	0.63 ±0.026
Our methods						
R2P2	$\mathbb{E}_{\hat{\mathbf{y}}}[\|\|\pi(\mathbf{y})-\pi(\hat{\mathbf{y}})\|\|_1] \cdot \ln q_\theta(\mathbf{y}\|\mathbf{x})$	7/100	8.86 ±0.188	9.26 ±0.194	2.29 ±0.022	0.58 ±0.010
R2P2	$\max_k \|\|\pi(\mathbf{y})-\pi(\hat{\mathbf{y}}^k)\|\|_1 \cdot \ln q_\theta(\mathbf{y}\|\mathbf{x})$	**1**/100	9.46 ±0.196	7.89 ±0.159	2.14 ±0.018	**0.55** ±0.011
Attention	$\alpha(\mathbf{x}) \cdot \ln q_\theta(\mathbf{y}_{\text{agent}}\|\mathbf{x}) + \ln q_\theta(\mathbf{y}_{\text{ego}}\|\mathbf{x})$	9/100	**14.36** ±0.217	**4.22** ±0.154	2.58 ±0.053	0.64 ±0.024
Oracle distribution		2/100	10.54 ±0.231	6.80 ±0.180	1.58 ±0.036	0.51 ±0.013

- By weighting prediction errors by their effect on downstream control, we can improve metrics we really care about: e.g., fewer collisions.
- This can decrease performance on tradition metrics like Average Displacement Error (ADE).

References

[1] Romina Abachi, Mohammad Ghavamzadeh, and Amir-massoud Farahmand. Policy-aware model learning for policy gradient methods. *arXiv preprint arXiv:2003.00030*, 2020.

[2] Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pages 463–474. PMLR, 2020.

[3] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3D tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019.

[4] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017.

[5] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. *arXiv preprint arXiv:2104.10133*, 2021.

[6] Amir-massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Artificial Intelligence and Statistics*, pages 1486–1494. PMLR, 2017.

[7] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. *arXiv preprint arXiv:2006.14480*, 2020.

[8] Joshua Joseph, Alborz Geramifard, John W Roberts, Jonathan P How, and Nicholas Roy. Reinforcement learning with misspecified model classes. In *International Conference on Robotics and Automation (ICRA)*, pages 939–946. IEEE, 2013.

[9] Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. *arXiv preprint arXiv:2002.04523*, 2020.

[10] Jean Mercat, Thomas Gilles, Nicole El Zoghby, Guillaume Sandou, Dominique Beauvois, and Guillermo Pita Gil. Multi-head attention for multi-modal joint vehicle motion forecasting. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9638–9644. IEEE, 2020.

[11] Nicholas Rhinehart, Kris Kitani, and Paul Vernaza. R2P2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *European Conference on Computer Vision*. Springer, 2018.