R2P2: A ReparameteRized Pushforward Policy for Diverse, Precise Generative Path Forecasting

Nicholas Rhinehart[1,2][0000−0003−4242−1236], Kris M. Kitani[1][0000−0002−9389−4060], and Paul Vernaza[2][0000−0002−2745−1894]

[1] Carnegie Mellon University, Pittsburgh PA 15213, USA
[2] NEC Labs America, Cupertino, CA 95014, USA

Abstract. We propose a method to forecast a vehicle's ego-motion as a distribution over spatiotemporal paths, conditioned on features (*e.g.*, from LIDAR and images) embedded in an overhead map. The method learns a policy inducing a distribution over simulated trajectories that is both "diverse" (produces most paths likely under the data) and "precise" (mostly produces paths likely under the data). This balance is achieved through minimization of a symmetrized cross-entropy between the distribution and demonstration data. By viewing the simulated-outcome distribution as the *pushforward* of a simple distribution under a simulation operator, we obtain expressions for the cross-entropy metrics that can be efficiently evaluated and differentiated, enabling stochastic-gradient optimization. We propose concrete policy architectures for this model, discuss our evaluation metrics relative to previously-used metrics, and demonstrate the superiority of our method relative to state-of-the-art methods in both the KITTI dataset and a similar but novel and larger real-world dataset explicitly designed for the vehicle forecasting domain.

Keywords: Trajectory Forecasting, Imitation Learning, Generative Modeling, Self-Driving Vehicles

1 Introduction

We consider forecasting a vehicle's trajectory (*i.e.*, predicting future paths). Forecasts can be used to foresee and avoid dangerous scenarios, plan safe paths, and model driver behavior. Context from the environment informs prediction, *e.g.* a map populated with features from imagery and LIDAR. We would like to learn a context-conditioned *distribution* over spatiotemporal trajectories to represent the many possible outcomes of the vehicle's future. With this distribution, we can perform inference tasks such as *sampling* a set of plausible paths, or *assigning a likelihood* to a particular observed path. Sampling suggests routes and visualizes the model; assigning likelihood helps measure the model's quality.

Our key motivation is to learn a trajectory forecasting model that is simultaneously "diverse"—covering all the modes of the data distribution—and "precise" in the sense that it rarely generates bad trajectories, such as trajectories that intersect obstacles. Fig. 1 contrasts a model trained to cover modes, versus

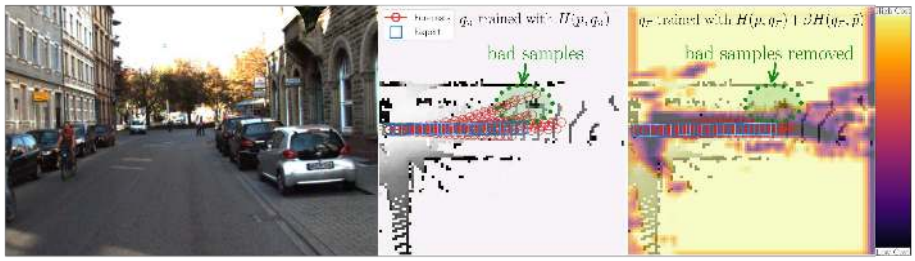


Fig. 1: *Left:* Natural image input. *Middle:* generated trajectories (red circles) and true, expert future (blue squares) overlaid on LIDAR map. *Right:* Generated trajectories respect approximate prior $\tilde{p}$, here a "cost function," overlaid as a heatmap. Making the expert paths likely corresponds to $\min_\pi H(p, q_\pi)$. Only producing likely paths corresponds to steering the trajectories away from unlikely territory via $\min_\pi H(q_\pi, \tilde{p})$. Doing both, *i.e.* producing most of the likely paths while mostly producing likely paths corresponds to $\min_\pi H(p, q_\pi) + \beta H(q_\pi, \tilde{p})$.

a model trained to cover modes *and* generate good samples, which generates fewer samples hitting perceived obstacles.

To achieve these ends, we propose learning a distribution over trajectories q_π that minimizes a symmetrized cross-entropy between q_π and the training data distribution, p. We represent q_π as a trajectory distribution induced by *rolling out* (simulating) a stochastic one-step *policy* π for T steps to produce a trajectory sample x. Denoting the scene context by ϕ, our objective can be written as

$$\min_\pi \underbrace{\mathbb{E}_{x\sim p} - \log q_\pi(x|\phi)}_{H(p,q_\pi)} + \beta \underbrace{\mathbb{E}_{x\sim q_\pi} - \log \tilde{p}(x|\phi)}_{H(q_\pi,\tilde{p})} \tag{1}$$

The two cross-entropy terms serve complementary purposes, as illustrated in Fig. 2: $H(p, q_\pi)$ encourages q_π to cover the modes of p, but fails to adequately penalize generating "low-quality" samples; $H(q_\pi, \tilde{p})$ encourages q_π to produce "high-quality" samples likely under an approximate data density $\tilde{p}$, but is insensitive to mode loss.

We advocate using the symmetrized cross-entropy metrics *for both training and evaluation* of trajectory forecasting methods. This is made feasible by viewing the distribution q_π as the *pushforward* of a base distribution under the function g_π that *rolls-out* (simulates) a stochastic policy π (see Fig. 3b). This idea (also known as the *reparameterization trick*, [22, 9]) enables optimization of model-sample quality metrics such as $H(q_\pi, \tilde{p})$ with SGD. Our representation also admits efficient accurate computation of $H(p, q_\pi)$, even when the policy is a very complex function of context and past state, such as a CNN.

We present the following novel contributions: 1) recognize and address the diversity-precision trade-off of generative forecasting models and formulating a symmetrized cross-entropy training objective to address it; 2) propose to train a policy to induce a roll-out distribution minimizing this objective; 3) use the

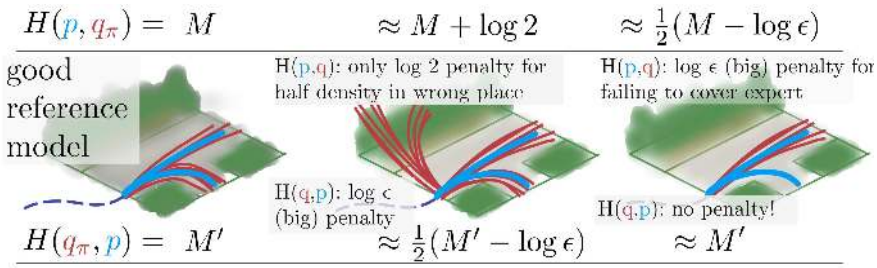


Fig. 2: Illustration of the complementarity of cross-entropies $H(p, q_\pi)$ (top) and $H(q_\pi, p)$ (bottom). Dashed lines show past vehicle path. Light blue lines delineate samples from the data (expert) distribution p. Samples from the model q_π are depicted as red lines. Green areas represent obstacles (areas with low p). The left figure shows cross-entropy values for a reference model. Other figures show poor models and their effects on each metric. ϵ is a very small nonnegative number.

pushforward parameterization to render inference and learning in this model efficient; 4) refine an existing deep imitation learning method (GAIL) based on our parameterization; 5) illuminate deficiencies of previously-used trajectory forecasting metrics; 6) outperform state-of-the-art forecasting and imitation learning methods, including our improvements to GAIL; 7) present CaliForecasting, a novel large scale dataset designed specifically for vehicle ego-motion forecasting.

2 Related Work

Trajectory Forecasting prior work spans two primary domains: trajectories of vehicles, and trajectories of people. The method of [26] predicts future trajectories of wide-receivers from surveillience video. In [50, 28, 5, 23] future pedestrian trajectories are predicted from surveillence video. Deterministic vehicle predictions are produced in [18], and deterministic pedestrian trajectories are produced in [3, 34, 30]. However, non-determinism is a key aspect of forecasting: the future is generally uncertain, with many plausible outcomes. While several approaches forecast distributions over trajectories [25, 12], global sample quality and likelihood have not been considered or measured, hindering performance evaluation.

Activity Forecasting is distinct from trajectory forecasting, as it predicts categorical activities. In [17, 24, 36, 35], future activities are predicted via classification-based approaches. In [33], a first-person camera wearer's future goals are forecasted with Inverse Reinforcement Learning (IRL). IRL has been applied to predict and control robot, taxi, and pedestrian behavior [31, 52, 23].

Imitation Learning can be used to frame our problem: learn a model to mimic an agent's behavior from a set of demonstrations [2]. One subtle difference is that in forecasting, we are not required to actually execute our plans in the real world. IRL is a form of imitation learning in which a reward function is learned to model demonstrated behavior. In the IRL method of [49], a cost map representation is

used to plan vehicle trajectories. However, no time-profile is represented in the predictions, preventing use of time-profiled metrics and modeling. GAIL [16, 27] is also a form of IRL, yet its adversarial framework and policy optimization are difficult to tune and lead to slow convergence. By adding the assumption of model dynamics, we derive a new differentiable GAIL training approach, supplanting the noisy, inefficient policy gradient search procedure. We show this easier-to-train approach achieves better performance in our domain.

Image Forecasting methods generate full image or video representations of predictions, endowing their samples with interpretability. In [44, 45, 43], unsupervised model are learned to generate sequences and representations of future images. In [46], surveillance image predictions of vehicles are formed by smoothing a patch across the image. [47] and [42] also predict future video frames with an intermediate pose prediction. In [10], predictions inform a robot's behavior, and in [40], policy representations for imitation and reinforcement learning are guided by a future observation forecasting objective. In [7], image boundaries are predicted. One drawback to image-based forecasting methods is difficulty in measurement, a drawback shared by many popular generative models.

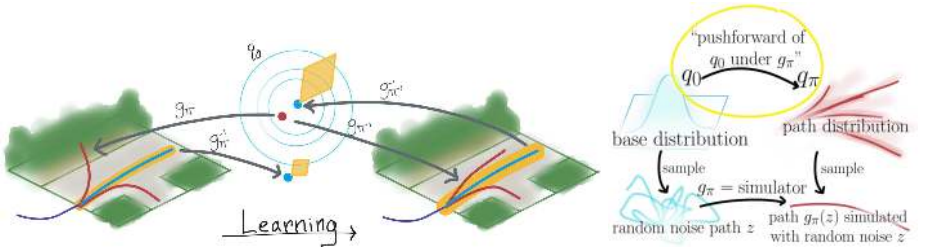
Generative models have surged in popularity [9, 14, 13, 16, 25, 44, 51]. However, one major difficulty is *performance evaluation*. Most popular models are quantified through heuristics that attempt to measures the "quality" of model samples [25]. In image generation, the Inception score is a popular heuristic [38]. These fail to measure the learned distribution's likelihood, the gold standard of evaluating probabilistic models. Notable exceptions include [9, 20], which also leverage invertible pushforward models to perform exact likelihood inference.

3 Approach

We approach the forecasting problem from an *imitation learning* perspective, learning a *policy* (state-to-action mapping) π that mimics the actions of an expert in varying contexts. We are given a set of training episodes (a short car path trajectory) $\{(x, \phi)_n\}_{n=1}^N$. Each episode $(x, \phi)_n$ has $x \in \mathbb{R}^{T\times 2}$ as a sequence of T two-dimensional future vehicle locations and ϕ as an associated set of side information. In our implementation, ϕ contains the past path of the car and a feature grid derived from LIDAR and semantic segmentation class scores. The grid is centered on the vehicle's position at $t = 0$ and is aligned with its heading.

Repeatedly applying the policy π from a start state with the context ϕ results in a distribution $q_\pi(x|\phi)$ over trajectories x, since our policy is stochastic. Similarly, the training set is drawn from a *data distribution* $p(x|\phi)$. We therefore train π so as to minimize a divergence between q_π and p. This divergence consists of a weighted combination of the cross-entropies $H(p, q_\pi)$ and $H(q_\pi, \tilde{p})$. The distribution $\tilde{p}$ is an approximation to p, which we assume cannot be evaluated. As discussed in Sec. 3.1, we might choose $\tilde{p}$ to be approximately uniformly distributed over non-obstacle regions. In the following, Φ denotes the distribution of ground-truth features:

$$\min_\pi \mathbb{E}_{\phi\sim\Phi}\left[-\mathbb{E}_{x\sim p(\cdot|\phi)} \log q_\pi(x|\phi) - \beta\mathbb{E}_{x\sim q_\pi(\cdot|\phi)} \log \tilde{p}(x|\phi)\right]. \tag{2}$$



(a) (b)

Fig. 3: (a) Consider making trajectories inside the yellow region on the road likelier by increasing $\log q_\pi(x)$ for the demonstration $x \sim p$ inside the region. This is achieved by making an infinitesimal region around $g_\pi^{-1}(x)$ more likely under q_0 by moving the region (yellow parallelogram, size proportional to $|\det J_{g_\pi}|^{-1}$) towards a mode of q_0 (here, the center of a Gaussian), and making the region bigger. Increasing $\log \tilde{p}(x)$ for some sample $x \sim q_\pi$ is equivalent to sampling a (red) point z from q_0 and adjusting π so as to increase $\log \tilde{p}(q_0(z))$. (b) Pushing forward a base distribution to a trajectory distribution.

The motivation for this objective is illustrated in Fig. 2. The two factors are complementary. $H(p, q_\pi)$ is intuitively similar to *recall* in binary classification, in that it is very sensitive to the model's ability to produce all of the examples in the dataset, but is relatively insensitive to whether the model produces examples that are unlikely under the data. $H(q_\pi, \tilde{p})$ is intuitively similar to *precision* in that it is very sensitive to whether the model produces samples likely under $\tilde{p}$, but is insensitive to q_π's likelihood to produce *all* samples in the dataset.

3.1 Pushforward distribution modeling

Optimizing Eq (2) presents at least two challenges: we must be able to evaluate $q_\pi(x|\phi)$ at arbitrary x in order to compute $H(p, q_\pi)$, and we must be able to differentiate the expression $\mathbb{E}_{x \sim q_\pi(\cdot|\phi)} \log \tilde{p}(x|\phi)$. We address these issues by constructing a learnable bijection, g_π between samples from q_π and samples from a simple noise distribution q_0, as illustrated in Fig. 3b; in our construction, the bijection is intepreted as a simulator mapping noise to simulated outcomes. This assumption allows us to evaluate the required expressions and derivatives via the change-of-variables formula and the *reparameterization trick*.

Specifically, let $g_\pi(z; \phi) : \mathbb{R}^{T \times 2} \to \mathbb{R}^{T \times 2}$ be a simulator mapping noise sequences $z \sim q_0$ and scene context ϕ to forecasted outcomes x. Then the distribution of forecasted outcomes $q_\pi(x|\phi)$ is fully determined by q_0 and g_π: this distribution is known as the *pushforward* of q_0 under g_π, as we are using g_π to "push forward" a distribution defined on the domain of g_π to one defined on its codomain. If g_π is differentiable and invertible, then q_π can be derived from the change-of-variables formula for multivariate integration:

$$q_\pi(x|\phi) = q_0\big(g_\pi^{-1}(x; \phi)\big)\big|\det J_{g_\pi}(g_\pi^{-1}(x; \phi))\big|^{-1}, \tag{3}$$

where $J_{g_\pi}(g_\pi^{-1}(x;\phi))$ is the Jacobian of g_π evaluated at $g_\pi^{-1}(x;\phi)$. This resolves both of the aforementioned issues: we can evaluate q_π and we can rewrite $\mathbb{E}_{x\sim q_\pi} \log \tilde{p}(x)$ as $\mathbb{E}_{z\sim q_0} \log \tilde{p}(g_\pi(z;\phi))$, since $g_\pi(z;\phi) \sim q_\pi$. The latter allows us to move derivatives wrt. π inside the expectation, as q_0 does not depend on π. Eq. (2) can then be rewritten as:

$$\min_\pi - \mathop{\mathbb{E}}_{\phi\sim\Phi} \mathop{\mathbb{E}}_{x\sim p(\cdot|\phi)} \log \frac{q_0(g_\pi^{-1}(x;\phi))}{\left|\det\, J_{g_\pi}(g_\pi^{-1}(x;\phi))\right|} - \beta \mathop{\mathbb{E}}_{z\sim q_0} \log \tilde{p}(g_\pi(z;\phi)|\phi) \tag{4}$$

Fig. 3a illustrates how this representation aids learning.

We note ours is not the only way to represent q_π and optimize Eq. (2). As long as q_π is analytically differentiable in the parameters, we may also apply REINFORCE [48] to obtain the required parameter derivatives. However, empirical evidence and some theoretical analysis suggests that the reparameterization-based gradient estimator typically yields lower-variance gradient estimates than REINFORCE [11]. This is consistent with the results we obtained in Sec. 4.

An invertible, differentiable simulator. In order to exploit the pushforward density formula (3), we must ensure g_π is invertible and differentiable. Inspired by [9, 21], we define g_π as an autoregressive map, representing the evolution of a controlled, discrete-time stochastic dynamical system with additive noise. Denoting $[x_1, \ldots, x_{t-1}]$ as $x_{1:t-1}$, and $[x_{1:t-1}, \phi]$ as ψ_t, the system is:

$$x_t \triangleq \mu_t^\pi(\psi_t;\theta) + \sigma_t^\pi(\psi_t;\theta) z_t, \tag{5}$$

where $\mu_t^\pi(\psi_t;\theta) \in \mathbb{R}^2$ and $\sigma_t^\pi(\psi_t;\theta) \in \mathbb{R}^{2\times 2}$ represent the stochastic one-step *policy*, and θ its parameters. The context, ϕ, is given in the form of a past trajectory $x_{\text{past}} = x_{-H_{\text{past}}+1:0} \in \mathbb{R}^{2H_{\text{past}}}$, and overhead feature map $M \in \mathbb{R}^{H_{\text{map}}\times W_{\text{map}}\times C}$: $\phi = (x_{\text{past}}, M)$. Note that the case $\sigma^\pi = 0$ would correspond to simply evolving the state by repeatedly applying μ^π—though this case is not allowed, as then g_π would not be invertible. However, as long as σ_t^π is invertible for all x, then g_π is invertible, and it is differentiable in x as long as μ^π and σ^π are differentiable in x. Since x_{τ_1} is not a function of x_{τ_2} for $\tau_1 < \tau_2$, the determinant of the Jacobian of this map is easily computed, because it is triangular (see supplement). Thus, we can easily compute terms in Eq. 4 via the following:

$$[g_\pi^{-1}(x)]_t = z_t = \sigma_t^\pi(\psi_t;\theta)^{-1}(x_t - \mu_t^\pi(\psi_t;\theta)) \tag{6}$$

$$\log\left|\det\, J_{g_\pi}(g_\pi^{-1}(x;\phi))\right| = \sum_t \log\left|\det\big(\sigma_t^\pi(\psi_t;\theta)\big)\right| \tag{7}$$

We note that q_π can also be computed via the chain rule of probability. For instance, if $z_t \sim$ is standard normal, then the marginal distributions are

$$q_\pi(x_t|\psi_t) = \mathcal{N}(x_t; \mu = \mu_t^\pi(\psi_t;\theta), \Sigma = \sigma_t^\pi(\psi_t;\theta)\sigma_t^\pi(\psi_t;\theta)^\top). \tag{8}$$

However, since it is still necessary to compute g_π in order to optimize $H(q_\pi, \tilde{p})$, we find it simplifies the implementation to compute q_π in terms of g_π.

Prior approximation of the data distribution. Evaluating $H(q_\pi, p)$ directly is unfortunately impossible, since we cannot evaluate the data distribution p's PDF. We therefore propose approximating it with a very simple density estimator $\tilde{p} \approx p$ trained independently and then fixed while training q_π. Simplicity reduces sample-induced variance in fitting $\tilde{p}$—crucial, because if $\tilde{p}$ severely underestimates p in some region R due to sampling error, then $H(q_\pi, \tilde{p})$ will erroneously assign a disproportionate penalty to samples from q_π landing in R.

We consider two options for $\tilde{p}$—first, simply using a kernel density estimator with a relatively large bandwidth. Since we have only one training sample per episode, this reduces to a single-kernel model. Choosing an isotropic Gaussian kernel, $H(q_\pi, \tilde{p})$ becomes $\mathbb{E}_{\hat{x}\sim q_\pi(\cdot|\phi)}\|x - \hat{x}\|^2/\gamma^2$, where (x, ϕ) constitutes an episode from the data. The net objective (2) in this case corresponds to $H(p, q_\pi)$ plus a mean squared distance penalty between model samples and data samples.

The second possibility is making an i.i.d. approximation; *i.e.*, parameterizing $\tilde{p}$ as $\tilde{p}(x \mid \phi) = \prod_t \tilde{p}_c(x_t \mid \phi)$. We proceed by discretizing x_t in a large finite region centered at the vehicle's start location; $\tilde{p}_c$ then corresponds to a categorical distribution with L classes representing the L possible locations. Training the i.i.d. model can then be reduced to training $\tilde{p}_c$ via logistic regression:

$$\min_{\tilde{p}} -\mathbb{E}_{x\sim p} \log \tilde{p}(x) = \max_{\theta} \mathbb{E}_{x\sim p} \sum_t -C_\theta(x_t, \phi) - \log \sum_{y=1}^{L} \exp -C_\theta(y, \phi), \quad (9)$$

where $C_\theta = -\log \tilde{p}_c$ can be thought of as a *spatial cost function* with parameters θ. We found it useful to decompose $C_\theta(y)$ as a sum $C^0_\theta(y) + C^1_\theta(y, \phi)$, where $C^0_\theta \in \mathbb{R}^L$ is thought of as a non-contextual location prior, and $C^1_\theta(y, \phi)$ has the form of a convolutional neural network acting on the spatial feature grid in ϕ and producing a grid of scores $\in \mathbb{R}^L$. Fig. 4 shows example learned $C^1_\theta(\cdot, \phi)$.

3.2 Policy modeling

We turn to designing learnable functions μ^π_t and σ^π_t. Across our three models, we use the following expansion: $\mu^\pi_t(\psi_t) = 2x_t - x_{t-1} + \hat{\mu}^\pi_t(\psi_t)$. The first terms correspond to a *constant velocity step* $(x_t + (x_t - x_{t-1}))$, and let us interpret $\hat{\mu}^\pi_t$ as a *deterministic acceleration*. Altogether, the update equation (Eq. 5) mimics Verlet integration [41], used to integrate Newton's equations of motion.

"Linear": The simplest model uses $\hat{\mu}^\pi_t, S_t$ linear in ψ_t:

$$\hat{\mu}^\pi_t(\psi_t) = Ah_t + b_0, \quad S_t(\psi_t) = Bh_t + b_1, \quad (10)$$

with $A \in \mathbb{R}^{2\times 2H}$, $h_t = x_{t-H:t-1} \in \mathbb{R}^{2H}$, $B \in \mathbb{R}^{4\times 2H}$, $b_i \in \mathbb{R}^{2H}$, and $S_t(\psi_t) \in \mathbb{R}^{2\times 2}$. To ensure positive-definiteness of σ^π_t, we use the matrix exponential [29]: $\sigma^\pi_t = \operatorname{expm}(S_t + S_t^\top)$, which we found to optimize more efficiently than $\sigma^\pi_t = S_t S_t^\top$.

"Field": The Linear model ignores M: it has no environment perception. We designed a CNN model that takes in M and outputs $O \in \mathbb{R}^{H_{\text{map}}\times W_{\text{map}}\times 6}$. The 6 channels in O are used to form the 6 components of μ^π_t and S_t in the following

(a) CaliForecasting Prior Examples (b) Kitti Prior Examples

Fig. 4: The prior penalizes positions corresponding to obstacles (white: high cost, black: low cost). The demonstrated expert trajectory is shown in each scene.

way. To ensure differentiability, the values in O are bilinearly interpolated at the current rollout position, x_t in the spatial dimensions (H_{map} and W_{map}) of O.

"RNN": The Linear and Field models reason with different contextual inputs: Linear uses the past, and CNN uses the feature map M. We developed a joint model to reason with both. M is passed through a CNN similar to Field's. The past is encoded with a GRU-RNN. Both featurizations inform a GRU-RNN that produces μ_t^π, S_t. See Figure 5 and the supplementary material for details.

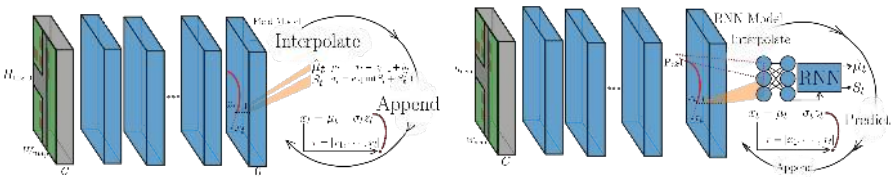


Fig. 5: RNN and CNN Policy models. The Field model produces a map of values to use for producing μ^π, σ^π through interpolation. The RNN model uses the same base as the Field model as well as information from the past trajectory to decode a featurized context representation and previous state to next μ^π, σ^π.

3.3 GAIL and Differentiable GAIL

As a deep generative approach to imitation learning, our method is comparable to Generative Adversarial Imitation Learning (GAIL [16]). GAIL is model-free: it is agnostic to model dynamics. However, this flexibility requires an expensive model-free policy gradient method, whereas the approach we have proposed is fully differentiable. The model-free approach is significantly disadvantaged in sample complexity [32, 19] in theory and practice. By assuming the dynamics are known and differentiable, as described in Sec. 3.1, we can also derive a version of GAIL that does not require model-free RL, since we can apply the reparameterization trick to differentiate the generator objective with respect to the policy parameters. A similar idea was explored for general imitation learning in [6]. We refer to this method as **R2P2 GAIL**. As our experiments show, R2P2 GAIL significantly outperforms standard GAIL, and our main model (R2P2) significantly outperforms and is easier to train than both GAIL and R2P2 GAIL.

4 Experiments

We implemented R2P2 and baselines with the primary aim of testing the following hypotheses. 1) The ability to exactly evaluate the model PDF should help R2P2 obtain better solutions than methods that do not use exact PDF inference (which includes GAIL). 2) The optimization of $H(p, q_\theta)$ should be correlated with the model's ability to cover the training data, in analogy to recall in binary classification. 3) Including $H(q_\theta, \tilde{p})$ in our objective should improve sample quality relative to methods without this term, as it serves a purpose analogous to precision in binary classification. 4) R2P2 GAIL will outperform GAIL through its more efficient optimization scheme.

4.1 The CaliForecasting Dataset

Current public datasets such as KITTI are suboptimal for the purpose of validating these hypotheses. KITTI is relatively small and was not designed with forecasting in mind. It contains relatively few episodes of subjectively interesting, nonlinear behavior. For this reason, we collected a novel dataset specifically designed for the ego-motion forecasting task, which we make public. The data is similar to KITTI in sensor modalities, but the data was collected so as to maximize the number of intersections, turning, and other subjectively interesting episodes. The data was collected with a sensor platform consisting of a Ford Transit Connect van with two Point Grey Flea3 cameras mounted on the roof in a wide-baseline configuration, in addition to a roof-mounted Velodyne VLP16 LIDAR unit and an IMU. The initial version of the dataset consists of three continuous driving sequences, each about one hour long, collected in mostly suburban areas of northern California (USA). The data was post-processed to produce a collection of episodes in the previously described format. The overhead feature map was populated by pretraining a semantic segmentation network [39], evaluating it on the sequences, correlating them with the LIDAR point cloud, and

binning the resulting semantic segmentation scores in addition to a height-above-ground plane feature. With a subsampling scheme of 2Hz, CaliForecasting consists of over 10,000 training, 1,200 validation and 1,200 testing examples. The Kitti splits, in comparison, are about 3,100 training, 140 validation, and slightly less than 500 test examples with a subsampling scheme of 1Hz.

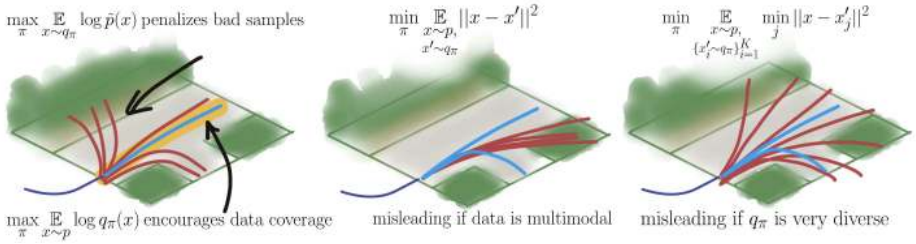


Fig. 6: Possible objectives and their attributes. $\min_\theta H(p, q_\theta)$ encourages data coverage, $\min_\theta H(q_\theta, \tilde{p})$ penalizes bad samples. Measuring mean squared error is misleading when the data is multimodal, and measuring mean squared error of the best sample fails to measure quality of samples far from the demonstrations.

4.2 Metrics and Baselines

Metrics Our primary metrics are the cross-entropy distribution metrics $H(p, q_\theta)$ and $H(q_\theta, \tilde{p})$. Note that $H(p, q_\theta)$ is lower-bounded by the entropy of p, $H(p)$, by Gibbs' inequality. Subtracting this quantity (computing KL) would be ideal; unfortunately, since $H(p)$ is unknown, we simply report $H(p, q_\theta)$. We also note that cross-entropy is *not* coordinate-invariant: we use path coordinates in an ego-centric frame that is a rotation and translation away from UTM coordinates (in meters) and report cross-entropy values for path distributions in this frame.

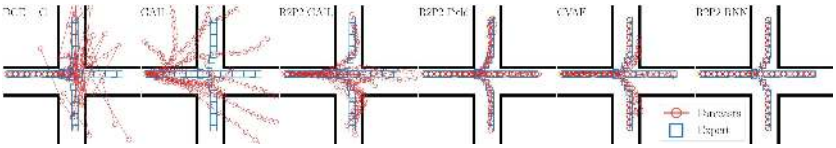
A subtle related issue is that $H(p, q_\theta)$ may be unbounded below since $H(p)$ may be arbitrarily negative. This phenomenon arises when the support of p is restricted to a submanifold—for example, if for $x \sim p$ and $x_1 - x_2 = b$, the distribution $q(x) \propto \exp(-\|x_1 - x_2 - b\|^2/\epsilon^2 + \|x\|^2/2)$ achieves arbitrarily low values of $H(p, q_\theta)$. We resolve this by slightly perturbing training and testing samples from p: *i.e.* instead of computing $H(p, q_\theta)$, we compute $-\mathbb{E}_{\eta\sim\mathcal{N}(0,\epsilon I)}\mathbb{E}_{x\sim p} \log q(x + \eta)$ for $\epsilon = 0.001$. This is lower-bounded by $H(\mathcal{N}(0, \epsilon I))$, which resolves the issue. See the supplement for more details.

We include two commonly used sample metrics [37, 3, 25, 15, 8], despite the shortcomings illustrated in Fig. 6. We measure the quality of the "best" sample from K samples from q_θ: $\hat{X}$, relative to the demonstrated sample x via $\mathbb{E}_{\hat{X}_k\sim q_\theta} \min_{\hat{x}\in\hat{X}_k} \|x - \hat{x}\|^2$ (known as "minMSD"). This metric fails to measure the quality of *all* of the samples, and thus can be exploited by an approach that predicts samples that are mostly poor. Additionally, we measure the mean

distance to the demonstration of all samples in $\hat{X}$: $\frac{1}{K}\sum_{k=1}^{K}\|x-\hat{x}_k\|^2$ (known as "meanMSD"). This metric is misleading if the data is multimodal, as the metric rewards predicting the mean, as opposed to covering multiple outcomes. *Due to the deficiencies of these common sample-based metrics for measuring the quality of multimodal predictions, we advocate supplementing sample-based metrics with the complementary cross-entropy metrics used in this work.*

Baselines. We construct a simple a unimodal baseline: given the context, the distribution of trajectories is given as a sequence of Gaussian distributions. This is called the **Gaussian Direct Cross-Entropy (DCE-G)**. As discussed in Section 3.3, we apply **Generative Adversarial Imitation Learning (GAIL)**, along with our modified GAIL framework, R2P2 GAIL. We constructed several variants of GAIL: with and without the (improved) Wasserstein-GAN [4, 14] parameterization, with and without our novel **R2P2 GAIL** formulation, and using the standard MLP discriminator, versus a CNN-based discriminator with a similar architecture to the Field model (details in supplementary). **Conditional Variational Autoencoders (CVAEs)** are a popular approach for modeling generative distributions conditioned on context. We follow the CVAE construction of [25] in our implementation. One key distinguishing factor is that CVAEs cannot perform *exact inference* by construction: given an arbitrary sample, a CVAE cannot produce a PDF value. Quantification of CVAE performance is thus required to be approximation-based, or sample-based. Our approaches are implemented in Tensorflow [1]. Architectural details are given in the supplement.

4.3 Cross Trimodal Experiments



Approach	DCE-G	GAIL	GAIL†	CVAE [25]	GAIL*	GAIL*†	Linear*	Field*	RNN*
Test $-H(p, q_\theta)$	-0.005	14.10	16.71	–	40.79	36.53	59.20	98.99	**120.7**
Test minMSD	56.90	28.29	21.54	0.001	28.61	1.264	28.54	1.179	**0.0006**

Fig. 7: Cross Trimodal Evaluation. *Top:* Qualitative results. *Bottom:* Quantitative results. A * indicates R2P2, and a † indicates using a WGAN Discriminator.

Our first set of experiments is designed to test the multimodal modeling capability of each approach in an easy domain. The contextual information is fixed – a single four-way intersection, along with three demonstrated outcomes: turning left, turning right, and going straight. Figure 7 shows qualitative and quantitative results. We see that several approaches fail to model multimodality well in this

Table 1: CaliForecasting and Kitti evaluation, $K = 12$

CaliForecasting Approach	Test $-H(p, q_\theta)$	Test minMSD	Test meanMSD	Test $-H(q_\theta, \tilde{p})$
DCE-G	-1.604 ± 0.02	4.953 ± 0.18	11.66 ± 0.27	-129.2 ± 0.43
GAIL-WG [16]	27.43 ± 0.03	9.117 ± 0.27	36.77 ± 2.50	-221.5 ± 2.40
CVAE [25]	$\approx 10.1 \pm 0.9$	1.680 ± 0.12	9.961 ± 0.25	-122.2 ± 0.48
R2P2 GAIL-WG	45.55 ± 0.07	5.529 ± 0.33	25.12 ± 0.80	-152.1 ± 1.00
R2P2 GAIL-WG CNN	43.55 ± 0.08	4.937 ± 0.26	26.59 ± 0.96	-154.3 ± 1.20
R2P2 Linear	64.02 ± 0.11	2.339 ± 0.14	10.51 ± 0.39	-144.5 ± 1.00
R2P2 Linear $\beta = 0.1$	61.57 ± 0.10	2.387 ± 0.13	11.27 ± 0.44	-134.1 ± 0.76
R2P2 Field	54.56 ± 0.11	2.171 ± 0.13	11.59 ± 0.39	-142.5 ± 0.75
R2P2 Field $\beta = 0.1$	53.88 ± 0.11	2.162 ± 0.11	10.87 ± 0.39	-132.8 ± 0.54
R2P2 RNN	$\mathbf{70.20} \pm 0.11$	$\mathbf{1.530} \pm 0.12$	11.25 ± 0.29	-125.0 ± 0.53
R2P2 RNN $\beta = 0.1$	66.89 ± 0.12	1.860 ± 0.14	10.68 ± 0.30	$\mathbf{-119.0} \pm 0.44$
R2P2 RNN $\gamma = 1.0$	65.12 ± 0.12	1.661 ± 0.11	$\mathbf{8.542} \pm 0.22$	-124.8 ± 0.48
Kitti Approach	Test $-H(p, q_\theta)$	Test minMSD	Test meanMSD	Test $-H(q_\theta, \tilde{p})$
DCE-G	-1.884 ± 0.03	6.217 ± 0.30	15.20 ± 0.62	-137.0 ± 0.72
GAIL-WG [16]	39.53 ± 0.11	5.517 ± 0.34	20.08 ± 2.00	-188.8 ± 1.76
CVAE [25]	$\approx 9.22 \pm 0.9$	1.436 ± 0.15	9.593 ± 0.52	-133.8 ± 1.21
R2P2 GAIL-WG	47.45 ± 0.16	4.062 ± 0.25	13.80 ± 1.10	-168.9 ± 1.50
R2P2 GAIL-WG CNN	42.49 ± 0.12	4.601 ± 0.30	19.87 ± 1.34	-164.2 ± 1.43
R2P2 Linear	62.39 ± 0.14	2.438 ± 0.16	16.16 ± 1.26	-163.4 ± 1.50
R2P2 Linear $\beta = 0.1$	63.82 ± 0.16	2.587 ± 0.15	28.33 ± 1.40	-151.1 ± 1.40
R2P2 Field	64.71 ± 0.18	1.717 ± 0.13	10.34 ± 0.59	-139.2 ± 1.10
R2P2 Field $\beta = 0.1$	62.79 ± 0.29	1.639 ± 0.13	10.92 ± 0.59	$\mathbf{-126.9} \pm 0.77$
R2P2 RNN	$\mathbf{67.70} \pm 0.20$	1.574 ± 0.15	10.46 ± 0.57	-131.6 ± 0.91
R2P2 RNN $\beta = 0.3$	65.80 ± 0.21	$\mathbf{1.282} \pm 0.09$	$\mathbf{9.352} \pm 0.55$	-130.8 ± 0.87

scenario. RNN. The models that can perform exact inference (all except CVAE) cover the modes with different success, as measured by Test $-H(p, q_\theta)$. We observe the models minimizing $H(p, q_\theta)$ cover the data well, supporting hypothesis 2 (coverage hypothesis), and outperform both GAIL approaches, supporting hypothesis 1 (exact inference hypothesis). We observe R2P2 GAIL outperforms GAIL in this scenario, supporting hypothesis 4 (optimization hypothesis). We also note the failure of DCE-G: its unimodal model is too restrictive for covering the diverse demonstrated behavior.

4.4 CaliForecasting Experiments and Kitti Experiments

We conducted larger-scale experiments designed to test our hypotheses. First, we trained $\tilde{p}$ on each dataset by the procedure described in Sec. 3.1. As discussed, our goal was to develop a simple model to minimize overfitting: we used a 3-layer Fully Convolutional NN. In the resulting spatial "cost" maps, we observe the model's ability to perceive obstacles in its assignment of low cost to on-road regions, and high-cost to clearly visible obstacles (*e.g* Fig. 4). We performed hyperparameter search for each method, and report the mean and its standard error of test set metrics corresponding to each method's best validation loss in Table 1. These results provide us with a rich set of observations. Of the three baselines, none catastrophically failed, with CVAE most often generating the cleanest samples. Across datasets and metrics, our approach achieves performance superior to the three baselines and our improved GAIL approach.

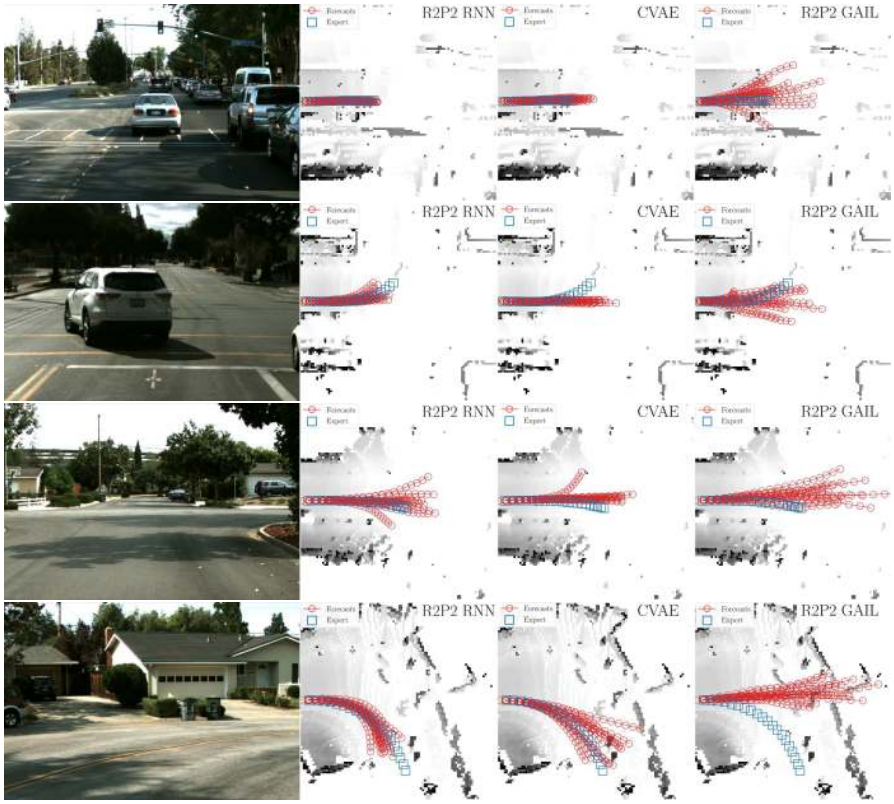


Fig. 9: CaliForecasting Results. Comparison of R2P2 RNN (middle-left), CVAE (middle-right), and R2P2 GAIL (right). Trajectory samples are overlaid on overhead LIDAR map, colored by height. *Bottom two rows:* Comparison of $\beta = 0$ (top) and $\beta = 0.1$ (bottom), overlaid on $\tilde{p}$ cost map. The cost map improves sample quality.

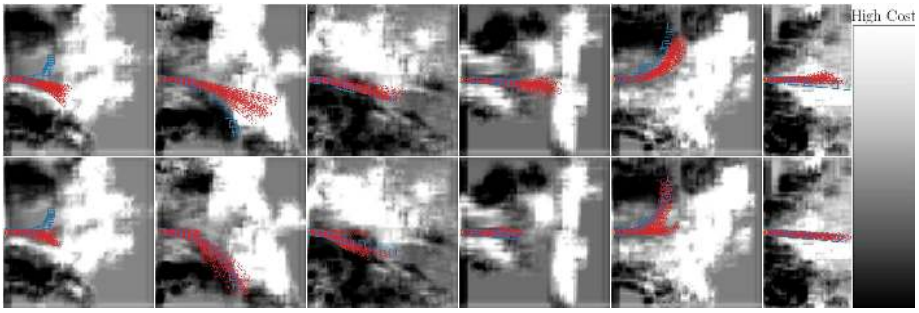


Fig. 10: Comparison of using β on CaliForecasting test data. *Top row:* With $\beta = 0$, some trajectories are forecasted into obvious obstacles. *Bottom row:* With $\beta \neq 0$, many forecasted trajectories do not hit obstacles.

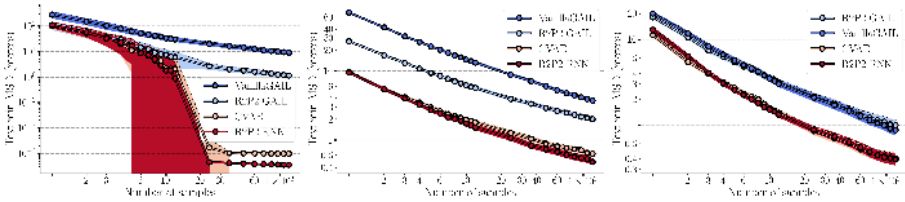


Fig. 11: Test $\min_k$ MSD vs. K on Cross, CaliForecasting, and Kitti.

By minimizing $H(p, q_\theta)$, our approach results in higher Test $-H(p, q_\theta)$ than all GAIL approaches, supporting the coverage and optimization hypotheses. We find that by incorporating our prior with nonzero β, hypothesis 3 is supported: our model architectures can improve the quality of its samples as measured by the Test $-H(q_\theta, \tilde{p})$. We observe that our GAIL optimization approach yields higher Test $-H(p, q_\theta)$, supporting hypothesis 4. We plot means and its standard error of the minMSD metrics as a function of K in Fig 11 for all 3 datasets.

We also find that qualitatively, our approach usually generates the best samples with diversity along multiple paths and precision in its tendency to avoid obstacles. Fig. 9 illustrates results on our dataset for our method, CVAE, and our improved GAIL approach. Fig. 10 illustrates qualitative examples for how incorporating nonzero β can improve sample quality.

5 Conclusions

This work has raised the previously under-appreciated issue of balancing diversity and precision in probabilistic trajectory forecasting. We have proposed a training a policy to induce a simulated-outcome distribution that minimizes a symmetrized cross-entropy objective. The key technical step that made this possible was a parameterizing the model distribution as the pushforward of a simple base distribution under the simulation operator. The relationship of this method to deep generative models was noted, and we showed that part of our full model enhances an existing deep imitation learning method. Empirically, we demonstrated that the pushforward parameterization enables reliable optimization of the objective, and that the optimized model has the desired characteristics of both covering the training data and generating high-quality samples. Finally, we introduced a novel large-scale, real-world dataset designed specifically for the vehicle ego-motion forecasting problem.

Acknowledgment. This work was sponsored in part by JST CREST (JPMJCR14E1) and IARPA (D17PC00340). Thanks to Anqi Liu, Wongun Choi, Kihyuk Sohn, and Manmohan Chandraker for insightful discussions.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: OSDI. vol. 16, pp. 265–283 (2016)
2. Abbeel, P., Ng, A.Y.: Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the twenty-first international conference on Machine learning. p. 1. ACM (2004)
3. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 961–971 (2016)
4. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
5. Ballan, L., Castaldo, F., Alahi, A., Palmieri, F., Savarese, S.: Knowledge transfer for scene-specific motion prediction. In: European Conference on Computer Vision. pp. 697–713. Springer (2016)
6. Baram, N., Anschel, O., Caspi, I., Mannor, S.: End-to-end differentiable adversarial imitation learning. In: International Conference on Machine Learning. pp. 390–399 (2017)
7. Bhattacharyya, A., Malinowski, M., Schiele, B., Fritz, M.: Long-term image boundary prediction. In: Thirty-Second AAAI Conference on Artificial Intelligence. AAAI (2017)
8. Bhattacharyya, A., Schiele, B., Fritz, M.: Accurate and diverse sampling of sequences based on a best of many sample objective. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8485–8493 (2018)
9. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. arXiv preprint arXiv:1605.08803 (2016)
10. Finn, C., Levine, S.: Deep visual foresight for planning robot motion. In: Robotics and Automation (ICRA), 2017 IEEE International Conference on. pp. 2786–2793. IEEE (2017)
11. Gal, Y.: Uncertainty in deep learning. Ph.D. thesis, University of Cambridge (2016)
12. Galceran, E., Cunningham, A.G., Eustice, R.M., Olson, E.: Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction. In: Robotics: Science and Systems XI, Sapienza University of Rome, Rome, Italy, July 13-17, 2015 (2015), http://www.roboticsproceedings.org/rss11/p43.html
13. Grover, A., Dhar, M., Ermon, S.: Flow-gan: Bridging implicit and prescribed learning in generative models. arXiv preprint arXiv:1705.08868 (2017)
14. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in Neural Information Processing Systems. pp. 5769–5779 (2017)
15. Gupta, A., Johnson, J.: Social gan: Socially acceptable trajectories with generative adversarial networks
16. Ho, J., Ermon, S.: Generative adversarial imitation learning. In: Advances in Neural Information Processing Systems. pp. 4565–4573 (2016)
17. Hoai, M., De la Torre, F.: Max-margin early event detectors. International Journal of Computer Vision **107**(2), 191–202 (2014)
18. Jain, A., Singh, A., Koppula, H.S., Soh, S., Saxena, A.: Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In: Robotics and Automation (ICRA), 2016 IEEE International Conference on. pp. 3118–3125. IEEE (2016)

19. Kakade, S.M., et al.: On the sample complexity of reinforcement learning. Ph.D. thesis (2003)
20. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. arXiv preprint arXiv:1807.03039 (2018)
21. Kingma, D.P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., Welling, M.: Improved variational inference with inverse autoregressive flow. In: Advances in Neural Information Processing Systems. pp. 4743–4751 (2016)
22. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
23. Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: European Conference on Computer Vision. pp. 201–214. Springer (2012)
24. Lan, T., Chen, T.C., Savarese, S.: A hierarchical representation for future action prediction. In: European Conference on Computer Vision. pp. 689–704. Springer (2014)
25. Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H., Chandraker, M.: Desire: Distant future prediction in dynamic scenes with interacting agents (2017)
26. Lee, N., Kitani, K.M.: Predicting wide receiver trajectories in american football. In: Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on. pp. 1–9. IEEE (2016)
27. Li, Y., Song, J., Ermon, S.: Infogail: Interpretable imitation learning from visual demonstrations. In: Advances in Neural Information Processing Systems. pp. 3815–3825 (2017)
28. Ma, W.C., Huang, D.A., Lee, N., Kitani, K.M.: Forecasting interactive dynamics of pedestrians with fictitious play. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. pp. 4636–4644. IEEE (2017)
29. Najfeld, I., Havel, T.F.: Derivatives of the matrix exponential and their computation. Advances in applied mathematics **16**(3), 321–375 (1995)
30. Park, H.S., Hwang, J.J., Niu, Y., Shi, J.: Egocentric future localization. In: CVPR. vol. 2, p. 4 (2016)
31. Ratliff, N.D., Bagnell, J.A., Zinkevich, M.A.: Maximum margin planning. In: Proceedings of the 23rd international conference on Machine learning. pp. 729–736. ACM (2006)
32. Recht, B.: The policy of truth. http://www.argmin.net/2018/02/20/reinforce/ (2018)
33. Rhinehart, N., Kitani, K.M.: First-person activity forecasting with online inverse reinforcement learning. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
34. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning social etiquette: Human trajectory understanding in crowded scenes. In: European conference on computer vision. pp. 549–565. Springer (2016)
35. Ryoo, M.S., Fuchs, T.J., Xia, L., Aggarwal, J.K., Matthies, L.H.: Robot-centric activity prediction from first-person videos: What will they do to me'. In: Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, HRI 2015, Portland, OR, USA, March 2-5, 2015. pp. 295–302 (2015). https://doi.org/10.1145/2696454.2696462, http://doi.acm.org/10.1145/2696454.2696462
36. Ryoo, M.S.: Human activity prediction: Early recognition of ongoing activities from streaming videos. In: Computer Vision (ICCV), 2011 IEEE International Conference on. pp. 1036–1043. IEEE (2011)
37. Sadeghian, A., Kosaraju, V., Gupta, A., Savarese, S., Alahi, A.: Trajnet: Towards a benchmark for human trajectory prediction. arXiv preprint (2018)

38. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems. pp. 2234–2242 (2016)
39. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., et al.: Going deeper with convolutions. Cvpr (2015)
40. Venkatraman, A., Rhinehart, N., Sun, W., Pinto, L., Hebert, M., Boots, B., Kitani, K., Bagnell, J.: Predictive-state decoders: Encoding the future into recurrent networks. In: Advances in Neural Information Processing Systems. pp. 1172–1183 (2017)
41. Verlet, L.: Computer" experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. Physical review **159**(1), 98 (1967)
42. Villegas, R., Yang, J., Zou, Y., Sohn, S., Lin, X., Lee, H.: Learning to generate long-term future via hierarchical prediction. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. pp. 3560–3569 (2017), http://proceedings.mlr.press/v70/villegas17a.html
43. Vondrick, C., Pirsiavash, H., Torralba, A.: Anticipating visual representations from unlabeled video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 98–106 (2016)
44. Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In: Advances In Neural Information Processing Systems. pp. 613–621 (2016)
45. Vondrick, C., Torralba, A.: Generating the future with adversarial transformers. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 2992–3000 (2017). https://doi.org/10.1109/CVPR.2017.319, https://doi.org/10.1109/CVPR.2017.319
46. Walker, J., Gupta, A., Hebert, M.: Patch to the future: Unsupervised visual prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3302–3309 (2014)
47. Walker, J., Marino, K., Gupta, A., Hebert, M.: The pose knows: Video forecasting by generating pose futures. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 3352–3361. IEEE (2017)
48. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. In: Reinforcement Learning, pp. 5–32. Springer (1992)
49. Wulfmeier, M., Rao, D., Wang, D.Z., Ondruska, P., Posner, I.: Large-scale cost function learning for path planning using deep inverse reinforcement learning. The International Journal of Robotics Research **36**(10), 1073–1087 (2017)
50. Xie, D., Todorovic, S., Zhu, S.C.: Inferring "dark matter" and "dark energy" from videos. In: Computer Vision (ICCV), 2013 IEEE International Conference on. pp. 2224–2231. IEEE (2013)
51. Zhu, J., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 2242–2251 (2017). https://doi.org/10.1109/ICCV.2017.244, https://doi.org/10.1109/ICCV.2017.244
52. Ziebart, B.D., Maas, A.L., Bagnell, J.A., Dey, A.K.: Maximum entropy inverse reinforcement learning. In: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008. pp. 1433–1438 (2008), http://www.aaai.org/Library/AAAI/2008/aaai08-227.php