Variational Adversarial Kernel Learned Imitation Learning

Fan Yang, Alina Vereshchaka, Yufan Zhou, Changyou Chen, Wen Dong
State University of New York at Buffalo
{fyang24, avereshc, yufanzho, changyou, wendong}@buffalo.edu

Abstract

Imitation learning refers to the problem where an agent learns to perform a task through observing and mimicking expert demonstrations, without knowledge of the cost function. State-of-the-art imitation learning algorithms reduce imitation learning to distribution-matching problems by minimizing some distance measures. However, the distance measure may not always provide informative signals for a policy update. To this end, we propose the variational adversarial kernel learned imitation learning (VAKLIL), which measures the distance using the maximum mean discrepancy with variational kernel learning. Our method optimizes over a large cost-function space and is sample efficient and robust to overfitting. We demonstrate the performance of our algorithm through benchmarking with four state-of-the-art imitation learning algorithms over five high-dimensional control tasks, and a complex transportation control task. Experimental results indicate that our algorithm significantly outperforms related algorithms in all scenarios.

Introduction

Reinforcement learning (RL) has made significant progress over a wide range of domains, ranging from Atari games (Mnih et al. 2016; 2015), traffic control (Li, Lv, and Wang 2016; Yang, Vereshchaka, and Dong 2018), to robotic control tasks (Andrychowicz et al. 2017; Kahn et al. 2018) through optimizing over predefined reward functions. However, designing an appropriate reward function for complex and not well-specified tasks is notoriously difficult (Li, Song, and Ermon 2017; Hadfield-Menell et al. 2017).

Rather than optimizing over predefined reward functions, imitation learning (IL) has the potential to close the gap by learning how to perform tasks through observing and mimicking expert demonstrations. Behavior cloning (BC) (Sammut and Webb 2011) is a classical imitation learning approach which directly learns the mapping from state to action through supervised learning. However, this method fails to learn a good policy in complex environments due to error compounding. Apprenticeship learning (AL) (Abbeel and Ng 2004) seeks to learn a policy that performs no worse than the expert policy. However, it assumes a restrictive class of cost functions – a linear combination of feature vectors. Generative adversarial imitation learning (GAIL) (Ho and Ermon 2016) reduces imitation learning to the problem of matching the state-action distribution of the learned policy to that of the expert policy. However, similar to generative adversarial networks (GAN) (Goodfellow et al. 2014), GAIL measures the distance between the distributions using the Jensen-Shannon (JS) divergence, which does not always provide informative signals to optimize the policy, leading to gradient vanishing.

Different approaches have been proposed to improve the training. Variational adversarial imitation learning (VAIL) (Peng et al. 2018) adopted a variational information bottleneck to introduce noise into the discriminator, which regulates the information flow and updates the policy using more useful features. (Kim and Park 2018) introduces generative moment matching imitation learning (GMMIL), where the cost function class is defined as the unit ball of a reproducing Hilbert kernel space (RKHS). The goal of GMMIL is to match the state-action distribution by minimizing the maximum mean discrepancy (MMD) with a pre-defined kernel.

To provide more informative signals for policy optimization, and to further improve the performance of a learned policy, we develop a new imitation learning algorithm, called the variational adversarial kernel learned imitation learning (VAKLIL). This algorithm follows the paradigm of formulating imitation learning as distribution matching, but with a novel distance measure defined as the MMD with variational kernel learning (MMD-VKL). VAKLIL has the following appealing properties: (1) It optimizes over a more expressive class of cost functions, which provides more informative and discriminative features for policy optimization. (2) It is sample efficient and is robust to overfitting.

The contributions of the paper are summarized as follows: (1) We develop an imitation learning algorithm with a more expressive class of cost functions, and reduce the imitation learning problem to state-action distribution matching. (2) We introduce a new distance measurement between distributions, the MMD-VKL, which is sample efficient and robust to overfitting. (3) We apply the new distance measurement to the proposed imitation-learning paradigm and develop VAKLIL, an imitation-learning algorithm that optimizes the policy with more informative signals, and which is sample efficient and robust. (4) To develop closed-form solutions, we provide two parameterizations of the kernel learning in VAKLIL, a Gaussian kernel, and a random Fourier kernel. (5) We analyze the theoretical properties of our method and compare the empirical performance of our algorithm with the aforementioned state-of-the-art algorithms such as AL, GAIL, VAIL, and GMMIL. Experimental results indicate that our algorithm significantly outperforms all algorithms in all scenarios.[1]

Background

MDP and Imitation Learning

Learning the policy through interacting with the environments can be modeled as a Markov decision process (MDP). Formally, a MDP is defined as a tuple $\langle S, A, P, c, \gamma \rangle$, where S represents the state space with $s_t \in S$ being the state at time t; A is the action space with $a_t \in A$ being the action taken at time t; P is the transition kernel of states $P(s_{t+1}|s_t, a_t)$; c is the cost function with $c(s_t, a_t)$ evaluating the immediate cost at time t; and $\gamma \in [0, 1)$ is a discount factor. The policy π is defined as a mapping from a state s_t to an action $a_t = \mu(s_t)$ (deterministic policy) or a conditional distribution parameterized by θ, *i.e.*, $\pi = p(a_t|s_t; \theta)$ (stochastic policy).

Given a cost function, the objective of policy learning can be formulated as minimizing the expected future costs: $J(c, \pi) = \mathbb{E}_\pi \left[\sum_t \gamma^t c(s_t, a_t)\right]$. For any policy π, there is one-to-one correspondence between the policy and its occupancy measure (Puterman 2014), which is also called the state-action visitation distribution $\rho_\pi(s, a) = \sum_t \gamma^t p(s_t = s, a_t = a)$. It can be shown that

$$J(c, \pi) = \mathbb{E}_{\rho_\pi}\left[c(s, a)\right] = \sum_{s,a} \rho_\pi(s, a) c(s, a) \ .$$

Recent approaches in IL (Ho and Ermon 2016) consider first applying inverse RL to find a cost function to best fit expert demonstrations, and then applying standard RL to optimize a policy under such a cost function. The objective is thus

$$\begin{aligned} \mathrm{IL}(\pi_E) \quad &= \mathrm{min}_\pi \mathrm{max}_{c \in C} \mathbb{E}_{\rho_\pi}[c(s, a)] - \mathbb{E}_{\rho_{\pi_E}}[c(s, a)] \\ &-\psi(c) - H(\pi) \end{aligned} \tag{1}$$

where $\psi(c)$ and $H(\pi)$ are the regularizations for the cost function and policy, respectively. This method can be viewed as minimizing the distance between the occupancy measure of the agent and that of the expert, where different IL algorithms use different measurement of the distance. For example, GAIL adopts the Jensen-Shannon divergence with an objective: $\mathrm{IL}(\pi_E) = \arg\ \mathrm{min}_\pi \mathrm{max}_D \mathbb{E}_{\rho_\pi}\left[\log\left(D(s, a)\right)\right] + \mathbb{E}_{\rho_{\pi_E}}\left[\log\left(1 - D(s, a)\right)\right] - \lambda H(\pi)$; and GMMIL uses MMD with a pre-defined kernel

$$\mathrm{IL}(\pi_E) = \mathrm{min}_\pi \mathrm{max}_{c \in C} M_k^2(\rho_\pi, \rho_{\pi_E}) \ .$$

Maximum Mean Discrepancy

Distinguishing two distributions by finite samples is known as Two-Sample Test in statistics. One way to achieve this is to evaluate via MMD (Gretton et al. 2012), which can be viewed as feature matching in a RKHS. More details about RKHS can be found in (Berlinet and Thomas-Agnan 2011) and Appendix. Given two distributions p and q, and a kernel k, let $\mathcal{H}_k$ denote the RKHS induced by kernel k. The square of MMD is defined as

$$\begin{aligned} M_k^2(P, Q) \quad &= \mathbb{E}_{x,x' \sim P}\left[k(x, x')\right] + \mathbb{E}_{y,y' \sim Q}\left[k(y, y')\right] \\ &-2\mathbb{E}_{x \sim P, y \sim Q}\left[k(x, y)\right] \end{aligned}$$

In practice the MMD is estimated with finite samples. Let samples $\{x_1, ..., x_n\} \sim p, \{y_1, ..., y_n\} \sim q$, the empirical estimation of $M_k^2(p, q)$ is $\tilde{M}_k^2(p, q) = \frac{1}{n^2}\sum_{i,j} k(x_i, x_j) + \frac{1}{n^2}\sum_{i,j} k(y_i, y_j) - \frac{2}{n^2}\sum_{i,j} k(x_i, y_j)$. Because of the sampling variance, $M_k^2(p, q)$ may be small even when p and q differs significantly. To have a stronger signal of the distance measurement, (Li et al. 2019) proposed an improvement of MMD by parameterizing the kernel with the following objective:

$$\begin{aligned} &\mathrm{max}_{\psi,\varphi} M_{\psi,\varphi}^2(P, Q) \\ &\triangleq \mathrm{max}_{\psi,\varphi} \mathbb{E}_{x,x' \sim P}\left[k_{\psi,\varphi}(x, x')\right] + \mathbb{E}_{y,y' \sim Q}\left[k_{\psi,\varphi}(y, y')\right] \\ &-2\mathbb{E}_{x \sim P, y \sim Q}\left[k_{\psi,\varphi}(x, y)\right] \end{aligned} \tag{2}$$

where $k_{\psi,\varphi}(x, x') = \mathbb{E}_\nu[e^{ih_\psi(\nu)^T(f_\varphi(x) - f_\varphi(x'))}]$, and $\nu \sim \mathcal{N}(0, I)$ is a standard Gaussian. Consequently, this solution consists of learning the base kernel $h_\psi(\nu)$ and the injected function $f_\varphi(x)$. Recent works showed that MMD GANs with a learned kernel via $\mathrm{min}_\theta \mathrm{max}_{\psi,\varphi} M_{\psi,\varphi}^2(p, q_\theta)$ achieves better performance than that of using a fixed kernel (Li et al. 2017; Bellemare et al. 2017; Bińkowski et al. 2018).

The Proposed Method

In this section, we develop an IL framework with a more expressive class of cost functions, and formulate IL as the problem of state-action distribution matching. We propose a new distance measure, denoted as MMD with variational kernel learning (MMD-VKL), and apply it to the proposed IL paradigm, resulting in variational adversarial kernel learned imitation learning (VAKLIL). We also propose two ways of parameterizing a kernel in VAKLIL, *e.g.*, via a Gaussian kernel and a random Fourier kernel. All proofs and derivations are postponed to the appendix.

Imitation Learning with a Broader Class of Cost Functions

The cost function in existing IL algorithms are defined in a restricted function class. For example, AL considers cost functions to be linear combinations of pre-defined features; VAIL and GAIL define a sophisticated penalty over the cost function; GMMIL defines the cost function as a unit ball of a pre-defined RKHS. Intuitively, with a wider range of cost-function class, more flexible policies could be learned.

[1] The appendix, codes and data used in this paper can be found in https://bit.ly/37rescf

To learn over a more expressive class of cost functions, we consider the cost function class over a series of RKHS spaces:

$$\mathcal{C}_{\mathcal{H}_{k_{\psi,\varphi}}} = \left\{ c(s,a) = \left\langle c, \phi_{k_{\psi,\varphi}}(s,a) \right\rangle_{\mathcal{H}_{k_{\psi,\varphi}}} \mid \forall \psi, \varphi \right\} \quad (3)$$

where $\phi_{k_{\psi,\varphi}}(s,a)$ is the feature vector, defined in a RKHS induced by kernel $k_{\psi,\varphi}$. Instead of defining the cost function in a fixed RKHS, our method spans the cost function over all RKHS spaces induced by a kernel $k_{\psi,\varphi}, \forall \psi, \varphi$.

Theorem 1 (Bochner's theorem (Rudin 1962))**.** *A continuous, real valued, symmetric and shift-invariant function k on $\mathbb{R}^d$ is a positive definite kernel if and only if there is a positive finite measure $p_k(\omega)$ such that $k(x - x') = \mathbb{E}_{\omega \sim p_k(\omega)}[e^{i\omega^T(x-x')}]$.*

Denote a state-action pair (s,a) as x, and further introduce a standard quadratic regularizer for the cost function $\psi(c) = \frac{1}{2} \parallel c \parallel^2_{\mathcal{H}_{k_{\psi,\varphi}}}$. The IL objective (1) becomes

$$\begin{aligned}
&\mathrm{IL}(\pi_E) \qquad (4)\\
=&\min\nolimits_\pi \max\nolimits_{c,\psi,\varphi} \langle c, \mathbb{E}_{x\sim\rho_\pi}\phi_{k_{\psi,\varphi}}(x) - \mathbb{E}_{x\sim\rho_{\pi_E}}\phi_{k_{\psi,\varphi}}(x)\rangle_{\mathcal{H}_{k_{\psi,\varphi}}}\\
&- \frac{1}{2} \parallel c \parallel^2_{\mathcal{H}_{k_{\psi,\varphi}}} - H(\pi)\\
=&\min\nolimits_\pi \max\nolimits_{\psi,\varphi} \mathbb{E}_{x,x'\sim\rho_\pi}\left[k_{\psi,\varphi}(x,x')\right] + \mathbb{E}_{x,x'\sim\rho_{\pi_E}}\left[k_{\psi,\varphi}(x,x')\right]\\
&- 2\mathbb{E}_{x\sim\rho_\pi,x'\sim\rho_{\pi_E}}\left[k_{\psi,\varphi}(x,x')\right] - H(\pi)\\
=&\min\nolimits_\pi \max\nolimits_{\psi,\varphi} M^2_{\psi,\varphi}(\rho_\pi, \rho_{\pi_E}) - H(\pi)
\end{aligned}$$

As a result, we reduce the standard IL with a cost function defined in (3) to the occupancy measure matching problem, using the evaluation metric of MMD with kernel learning (MMD-KL). MMD-KL is an improvement over MMDs that simultaneously learns a kernel that maximizes the MMD. Compared to GMMIL, which measures the distance using MMD with a fixed kernel, MMD-KL is shown to produce larger distance measurement between ρ_π and ρ_{π_E}, hence a stronger signal to train the policy π. For more details, please see (Fukumizu et al. 2009).

The objective (4) is optimized by

- (1) Learning the kernel $k_{\psi,\varphi}$ to obtain the best measure of the distance $\max_{\psi,\varphi} M^2_{\psi,\varphi}(\rho_\pi, \rho_{\pi_E})$;
- (2) Optimizing the policy π to minimize the distance.

In kernel learning, we fix the policy π and update the parameters ψ, φ using gradient ascent

$$\begin{aligned}
\psi &= \psi + \nabla_\psi M^2_{\psi,\varphi}(\rho_\pi, \rho_{\pi_E})\\
\varphi &= \varphi + \nabla_\varphi M^2_{\psi,\varphi}(\rho_\pi, \rho_{\pi_E})
\end{aligned}$$

until converged. In policy optimization, we firstly obtain the optimal cost function through

$$c^* = \mathbb{E}_{x\sim\rho_\pi}\phi_{k_{\psi,\varphi}}(x) - \mathbb{E}_{x\sim\rho_{\pi_E}}\phi_{k_{\psi,\varphi}}(x) ,$$

then optimize the policy according to

$$\arg\min\nolimits_\pi \mathbb{E}_{x\sim\rho_\pi}\left[\hat{c}(x)\right] - H(\pi) ,$$

which can be solved using any maximum entropy RL algorithms with cost $\hat{c}(x) = \mathbb{E}_{x'\sim\rho_\pi}\left[k_{\psi,\varphi}(x,x')\right] - \mathbb{E}_{x'\sim\rho_{\pi_E}}\left[k_{\psi,\varphi}(x,x')\right]$. In practice, we evaluate the expectation with samples, *e.g.*, given two sets of samples $\{x_i\}_{i=1}^N \sim \rho_\pi$ and $\{y_i\}_{i=1}^N \sim \rho_{\pi_E}$, $M^2_{\psi,\varphi}(\rho_\pi, \rho_{\pi_E})$ is approximated as

$$\begin{aligned}
\tilde{M}^2_{\psi,\varphi}(\rho_\pi, \rho_{\pi_E}) = \frac{1}{N^2}\sum_{i,j} k_{\psi,\varphi}(x_i,x_j) + \frac{1}{N^2}\sum_{i,j} k_{\psi,\varphi}(y_i,y_j)\\
- \frac{2}{N^2}\sum_{i,j} k_{\psi,\varphi}(x_i,y_j) .
\end{aligned}$$

Similarly, the cost is calculated as $\tilde{c}(x) = \frac{1}{N}\sum_i k_{\psi,\varphi}(x,x_i) - \frac{1}{N}\sum_i k_{\psi,\varphi}(x,y_i)$.

Sample Efficient and Robust Imitation Learning

The above procedure of optimizing $M^2_{\psi,\varphi}(\rho_\pi, \rho_{\pi_E})$ with kernel learning involves repeatedly (1) sampling data from distributions ρ_π, ρ_{π_E}, and (2) updating kernel parameters ψ, φ with gradient ascent (Li et al. 2017). However, there is a tradeoff in kernel learning. On the one hand, collecting trajectory samples through interacting with an environment is typically time consuming and expensive (Jeon, Seo, and Kim 2018). One way to alleviate this is to reuse the samples. On the other hand, if samples are reused, repeated updating the kernel $k_{\psi,\varphi}$ with the same samples would lead to overfitting. As a result, the optimal cost function induced from the learned kernel would not provide good signals to learn a policy. To provide a sample efficient training paradigm and to impose a regularization to penalize overfitting, we propose a novel general kernel-learning scheme within the IL framework.

MMD with variational kernel learning We consider a general kernel learning problem where one learns a kernel $k_{\psi,\varphi}$ to maximize $\max_{\psi,\varphi} M^2_{\psi,\varphi}(p,q)$ between two distributions p and q. To avoid overfitting, inspired by the information bottleneck (Peng et al. 2018; Alemi et al. 2016), we constrain the information flow through the kernel. Specifically, we introduce an encoder E that maps a sample x to a stochastic encoding $z \sim E(z \mid x)$, and the goal is to learn an encoding z that is maximumly informative about the target $\max_{\psi,\varphi} M^2_{\psi,\varphi}(p,q)$, and meanwhile being maximumly compressive about the input samples x. The proposed kernel learning objective is thus

$$\max\nolimits_{E,\psi,\varphi} M^2_{E,\psi,\varphi}(p,q) - \beta I(X,Z) \qquad (5)$$

We refer to (5) the MMD with variational kernel learning, where

$$\begin{aligned}
M^2_{E,\psi,\varphi}(p,q) = \mathbb{E}_{x,x'\sim p}\mathbb{E}_{z\sim E(z|x),z'\sim E(z'|x')}[k_{\psi,\varphi}(z,z')]\\
+ \mathbb{E}_{y,y'\sim q}\mathbb{E}_{z\sim E(z|y),z'\sim E(z'|y')}[k_{\psi,\varphi}(z,z')]\\
- 2\mathbb{E}_{x\sim p,y\sim q}\mathbb{E}_{z\sim E(z|x),z'\sim E(z'|y)}[k_{\psi,\varphi}(z,z')] ,
\end{aligned}$$

and $I(X,Z)$ is the mutual information between the original samples x and the encoding z; and $\beta \geq 0$ is a coefficient. Intuitively, the first term $M^2_{E,\psi,\varphi}(p,q)$ encourages z to maximize the distance between p and q, and the second term $\beta I(X,Z)$ encourages z to ignore as much information flowing through as possible, where β controls the tradeoff. When trained adversarially, this will make the encoding z to only keep the most useful and common pattern from x that can be used to distinguish p and q. Essentially, it learns an encoding z as a minimal sufficient statistics of the input samples x to maximize the distance between p and q, thus is robust to overfitting. In practice, we define the encoder as

$$E(z \mid x) = \mathcal{N}(\mu_E(x), \Sigma_E(x)) \ .$$

An encoding z is sampled using the reparameterization trick $z = \mu_E(x) + \epsilon \Sigma_E^{1/2}(x)$, where $\epsilon \sim \mathcal{N}(0, I)$ is a standard Gaussian.

It is worth noting that not all MMD with kernel learning can provide a good measurement of the distance. (Li et al. 2019; Arbel et al. 2018) showed that the MMD with kernel learning needs to be weak (i.e. $p \overset{D}{\to} q \Leftrightarrow \max_{\psi,\varphi} M^2_{\psi,\varphi}(p,q) \to 0$) to provide informative indicator of the distance. We prove that the proposed MMD with variational kernel learning is weak over the encodings z under mild assumptions used in (Li et al. 2017; Arjovsky, Chintala, and Bottou 2017).

Theorem 2. *Assume 1) the function $f_\varphi(x)$ is bounded and has a common Lipschitz constant $\sup_\varphi \parallel f_\varphi(x) \parallel_L \leq L_\varphi < \infty$; 2) the variance of function $h_\psi(\nu)$ is bounded $\mathbb{E}_\nu \left[\parallel h_\psi(\nu) \parallel^2\right] < \infty$; 3) and the kernel is bounded $sup_x k_{\psi,\varphi}(x,x) < \infty$. Let p' and q' be the encoding distributions, i.e., $p'(z) = E(z \mid x), x \sim p$, and $q'(z) = E(z \mid x), x \sim q$, then $max_{E,\psi,\varphi} M^2_{E,\psi,\varphi}(p,q)$ is continuous in the weak topology for the encodings z:*

$$max_{E,\psi,\varphi} M^2_{E,\psi,\varphi}(p,q) \to 0 \Longleftrightarrow p' \overset{D}{\to} q'$$

Theorem 2 shows that $\max_{E,\psi,\varphi} M^2_{E,\psi,\varphi}(p,q)$ provides a informative indicator of the distance in the encoding z. In practice, we incorporate Lipschitz constraints $\parallel E \parallel_L \leq 1, \parallel f_\phi \parallel_L \leq 1$, and penalize $\mathbb{E}_\nu \left[\parallel h_\psi(\nu) \parallel^2\right]$ to accommodate the assumptions in Theorem 2. Let $\lambda_h \geq 0$ be the penalizing coefficient, the MMD with variational kernel learning is formulated as

$$\begin{aligned} &\max_{E,\psi,\varphi:\|E\|_L \leq 1, \|f_\varphi\|_L \leq 1} M^2_{E,\psi,\varphi}(\rho_\pi, \rho_{\pi_E}) \\ &-\beta I(X,Z) - \lambda_h \mathbb{E}_\nu \left[\parallel h_\psi(\nu) \parallel^2\right] \end{aligned} \tag{6}$$

Variational adversarial kernel learned imitation learning Incorporating (6) into the IL objective (4) gives the objective of the proposed VAKLIL framework:

$$\begin{aligned} \mathrm{IL}(\pi_E) = \min_\pi \max_{E,\psi,\varphi:\|E\|_L \leq 1, \|f_\varphi\|_L \leq 1} & M^2_{E,\psi,\varphi}(\rho_\pi, \rho_{\pi_E}) \\ - \beta I(X,Z) - \lambda_h \mathbb{E}_\nu \left[\parallel h_\psi(\nu) \parallel^2\right] - H(\pi) & \end{aligned} \tag{7}$$

Here, the terms $I(X,Z), \mathbb{E}_\nu \left[\parallel h_\psi(\nu) \parallel^2\right], \parallel E \parallel_L \leq 1, \parallel f_\varphi \parallel_L \leq 1$ can be viewed as the regularizations for the encoding E and the functions h_ψ, f_φ.

Theorem 3. *Solving the IL problem described by* (7) *is equivalent to solving a regularized IL problem with a cost function defined over the stochastic encoding $z \sim \mathbb{E}(z \mid s,a)$, with the cost function class being*

$$\begin{aligned} \mathcal{C}_{\mathcal{H}_{k_{\psi,\varphi}}} = \Bigg\{ & c(s,a) = \mathbb{E}_{z \sim E(z|s,a)} \left[c(z)\right] \\ & = \left\langle c, \mathbb{E}_{z \sim E(z|s,a)} \phi_{k_{\psi,\varphi}}(z) \right\rangle_{\mathcal{H}_{k_{\psi,\varphi}}} \mid \forall \psi, \varphi \Bigg\} , \end{aligned}$$

and the regularizations being $I(X,Z) \leq I_c, \mathbb{E}_\nu[\parallel h_\psi(\nu) \parallel^2] < \infty, \parallel E \parallel_L \leq 1, \parallel f_\varphi \parallel_L \leq 1$.

Theorem 3 draws a connection between (7) and the cost function class $\mathcal{C}_{\mathcal{H}_{k_{\psi,\varphi}}}$, where the cost function class spans over a class of RKHS induced by all shift-invariant kernels $k_{\psi,\varphi}$, and the cost function is a stochastic function defined over the encoding z. It implies that solving a regularized imitation learning problem with a stochastic cost function given in Theorem 3 would, as discussed in previous sections, reduce overfitting of the kernel learning. Furthermore, it could provide better signals for policy optimization.

We solve (7) by alternating between kernel learning with parameters (E, ψ, φ) and policy optimization. The algorithm is given as Algorithm 1, and a flow diagram is shown in Figure 1. More detailed descriptions of the optimization are provided in the Appendix. Denote a state-action pair (s,a) as x, we again adopt a Gaussian encoder $E(z \mid x) = \mathcal{N}(\mu_E(x), \Sigma_E(x))$ and use the reparameterization trick to reparameterize the encoding. In the algorithm, spectral normalization (Miyato et al. 2018) is used to satisfy the Lipschitz constraints $\parallel E \parallel_L \leq 1, \parallel f_\varphi \parallel_L \leq 1$, and samples estimations are used to approximate expectations in the MMD.

Algorithm 1: Variational adversarial kernel learned imitation learning

```
Input  : Expert dataset of trajectories
         $D_{\pi_E} = \left\{\left(s_i^E, a_i^E\right)\right\}_{i=1}^N$, initial policy $\pi$, initial
         kernel parameters $E, \psi, \varphi$, coefficient $\beta, \lambda_h$
Output : Learned policy $\pi$
for $iter = 0, 1, ...$ do
    Sample trajectories $D_\pi = \{(s_i, a_i)\}_{i=1}^M$ by
      executing $\pi$, sample $M$ state-action pairs from
      $D_{\pi_E}$, labeling them $i = 1, .., M$: $\left\{\left(s_i^E, a_i^E\right)\right\}_{i=1}^M$
    for $ik = 0, 1, ...$ do
        Update kernel parameters $E, \psi, \varphi$ with gradient
          ascent to maximize the objective
          $J = M^2_{E,\psi,\varphi}(\rho_\pi, \rho_{\pi_E}) - \beta I(X,Z) -$
          $\lambda_h \mathbb{E}_\nu \left[\parallel h_\psi(\nu) \parallel^2\right]$
    end
    Compute the cost function $\hat{c}(s,a) =$
      $\mathbb{E}_{s',a' \sim \rho_\pi, z' \sim E(z'|s',a')} \left[k_{\psi,\varphi}(\mu_E(s,a), z')\right] -$
      $\mathbb{E}_{s'',a'' \sim \rho_{\pi_E}, z'' \sim E(z''|s'',a'')} \left[k_{\psi,\varphi}(\mu_E(s,a), z'')\right]$
    Update policy $\pi$ using TRPO with the above cost
      function.
end
```

Theorem 4. *Let the policy π_θ be parameterized by θ, and $\epsilon \sim \mathcal{N}(0, I)$. The gradient of the policy optimization in VAKLIL has the form*

$$\begin{aligned} &\nabla_\theta \left(\mathbb{E}_{x \sim \rho_{\pi_\theta}} \left[\hat{c}(x)\right] - H(\pi_\theta)\right) \\ =&\nabla_\theta \mathbb{E}_{x \sim \rho_{\pi_\theta}, x' \sim \rho_{\pi_\theta}, \epsilon} \Big\langle \phi_{k_{\psi,\varphi}}(\mu_E(x') + \epsilon \Sigma_E^{1/2}(x')), \\ &\quad \phi_{k_{\psi,\varphi}}(\mu_E(x)) \Big\rangle_{\mathcal{H}_{k_{\psi,\varphi}}} \\ &- \nabla_\theta \mathbb{E}_{x \sim \rho_{\pi_\theta}, x'' \sim \rho_{\pi_E}, \epsilon} \Big\langle \phi_{k_{\psi,\varphi}}(\mu_E(x'') + \epsilon \Sigma_E^{1/2}(x'')), \end{aligned}$$

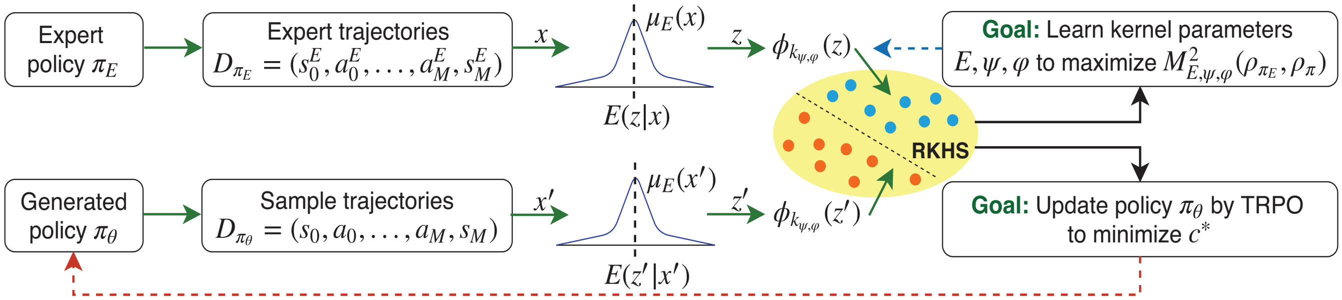
Expert policy π_E
Expert trajectories $D_{\pi_E} = (s_0^E, a_0^E, \dots, a_M^E, s_M^E)$
x
$\mu_E(x)$
$E(z|x)$
z
$\phi_{k_{\psi,\varphi}}(z)$
Goal: Learn kernel parameters E, ψ, φ to maximize $M^2_{E,\psi,\varphi}(\rho_{\pi_E}, \rho_\pi)$
RKHS
Generated policy π_θ
Sample trajectories $D_{\pi_\theta} = (s_0, a_0, \dots, a_M, s_M)$
x'
$\mu_E(x')$
$E(z'|x')$
z'
$\phi_{k_{\psi,\varphi}}(z')$
Goal: Update policy π_θ by TRPO to minimize c^*

Figure 1: VAKLIL flow diagram. Given expert data D_{π_E}, a VAKLIL learns a policy π_θ through iteratively: (1) Collecting sample trajectories D_{π_θ} from executing policy π_θ; (2) Applying a stochastic encoding $E(z \mid x)$ to the samples, and then projecting the encodings z to a feature vector $\phi_{k_{\psi,\varphi}}(z)$ in a RKHS; (3) Kernel learning with parameters (E, ψ, φ) to maximize the distance $M^2_{E,\psi,\varphi}(\rho_\pi, \rho_{\pi_E})$ between the generated samples and that of expert in the RKHS; (4) Updating the policy π_θ by TRPO to minimize the cost function c^* induced by the learned kernel $k_{\psi,\varphi}$.

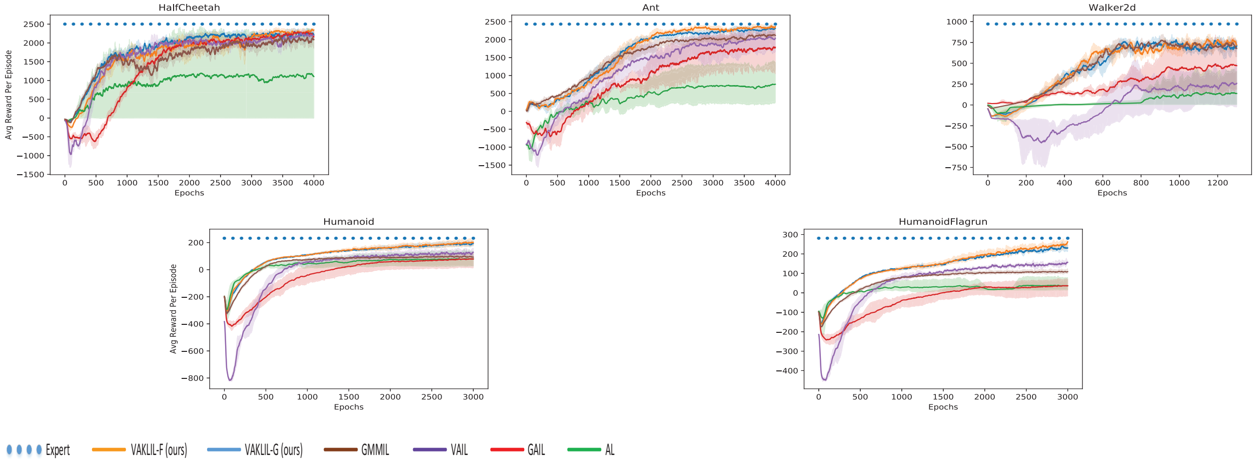

Figure 2: Learning curves of our algorithm, VAKLIL, and other algorithms.

$$\phi_{k_{\psi,\varphi}}(\mu_E(x))\Big\rangle_{\mathcal{H}_{k_{\psi,\varphi}}}$$

$$-\nabla_\theta H(\pi_\theta)\,.$$

Theorem 4 indicates that the gradient of the policy drives to maximize a inner product, $\left\langle \phi_{k_{\psi,\varphi}}(\mu_E(x'') + \epsilon\Sigma_E^{1/2}(x'')), \phi_{k_{\psi,\varphi}}(\mu_E(x)) \right\rangle_{\mathcal{H}_{k_{\psi,\varphi}}}$, in a RKHS. This drives the generated samples $x \sim \rho_{\pi_\theta}$ towards the expert samples $x'' \sim \rho_{\pi_E}$ through matching the features, $\phi_{k_{\psi,\varphi}}(\mu_E(x))$, of the generated samples to that of the expert samples in a RKHS, which is expressed as $\phi_{k_{\psi,\varphi}}(\mu_E(x'') + \epsilon\Sigma_E^{1/2}(x''))$. Here $\mu_E(x'') + \epsilon\Sigma_E^{1/2}(x'')$ is the stochastic (Gaussian) output from the encoder. Meanwhile, the gradient also drives to minimize the inner product

$$\left\langle \phi_{k_{\psi,\varphi}}(\mu_E(x') + \epsilon\Sigma_E^{1/2}(x')), \phi_{k_{\psi,\varphi}}(\mu_E(x)) \right\rangle_{\mathcal{H}_{k_{\psi,\varphi}}},$$

which pushes the generated samples $x \sim \rho_{\pi_\theta}$ far away from other generated samples $x' \sim \rho_{\pi_\theta}$ through maximizing the distance between the features of the generated samples $\phi_{k_{\psi,\varphi}}(\mu_E(x))$ to that of the other generated samples repersented as $\phi_{k_{\psi,\varphi}}(\mu_E(x') + \epsilon\Sigma_E^{1/2}(x'))$.

Kernel Parameterizations

The kernel in our algorithm takes the form $k_{\psi,\varphi}(x, x') = \mathbb{E}_\nu\left[e^{ih_\psi(\nu)^T(f_\varphi(x) - f_\varphi(x'))}\right]$, which involves complex exponential. To provide closed form kernel evaluations, we propose two ways of parameterizing the kernel $k_{\psi,\varphi}(x, x')$: a Gaussian kernel, and a random Fourier kernel.

Theorem 5. *Let $p_k(\omega) = p(h_\psi(\nu))$ be the spectral distribution.*

- *1) If $p_k(\omega)$ is fixed to be $(2\pi)^{-\frac{D}{2}} e^{-\frac{\|\omega\|^2}{2}}$, the kernel $k_{\psi,\varphi}(x, x')$ becomes a Gaussian kernel $k_\varphi(x, x') =$*

Environment	HalfCheetah	Ant	Walker2D	Humanoid	HumanoidFlagrun
Expert	2501.5±14.3	2433.9±17.2	971.6±1.2	232.2±1.6	281.0±2.1
VAKLIL-G	2234.0±313.4	2302.2±124.0	711.4±234.9	188.2±33.7	217.6±23.2
VAKLIL-F	2339.5±153.5	2348.8±136.4	735.2±154.7	198.7 ±40.9	262.8±31.6
AL	1145.3±1155.0	752.4±701.0	138.4±306.8	79.2±69.0	36.7±30.6
GAIL	2214.6±196.6	1780.0±597.0	476.8±326.7	78.7±64.2	34.2±37.0
VAIL	2197.8±113.5	2015.6±280.2	257.1±236.3	120.2±40.3	158.6±31.5
GMMIL	2099.1±366.1	2127.1±106.7	706.0±184.9	94.6±16.2	111.9±11.7

Table 1: The mean and standard deviation of the reward of the learned policy for different algorithms.

$e^{-\frac{(f_\varphi(x)-f_\varphi(x'))^2}{2}}$.

- *2) If the kernel is estimated using random samples $\omega_i \sim p_k(\omega)$ and the complex exponential is replaced with cosine, then it becomes a random Fourier kernel $k_{\psi,\varphi}(x, x') = \kappa(x)\kappa(x')$, where $\kappa(x) = \sqrt{\frac{1}{D}}\Big[cos\big(h_\psi(\nu_1)^T f_\varphi(x) + b_1\big), ..., cos\big(h_\psi(\nu_D)^T f_\varphi(x) + b_D\big)\Big]'$.*

Theorem 5 draws a connection among a shift-invariant kernel in the general form, a Gaussian kernel, and a random Fourier kernel. Kernel learning with a Gaussian kernel is equivalent as learning a shift-invariant kernel with injected function $f_\varphi(x)$ while fixing the base kernel $h_\psi(\nu)$. A random Fourier kernel uses an empirical estimation of the spectral distribution with samples $\omega \sim p_k(\omega)$. Kernel learning with a random Fourier kernel implies learning both the injected function $f_\varphi(x)$ and the base kernel $h_\psi(\nu)$. We will demonstrate the performance of VAKLIL with these two forms of kernel parameterizations in the experiments.

Connections to Existing Works

Apprenticeship learning (AL) (Abbeel and Ng 2004) defines a cost function as a linear combination of pre-defined finite dimensional features $c(s, a) = w\phi(s, a)$. Optimizing the policy under such cost function is equivalent to feature matching. In our method, we use a more expressive class of cost functions defined in a series of RKHS, which implies infinite-dimension feature matching.

GAIL (Ho and Ermon 2016) and VAIL (Peng et al. 2018) match the occupancy measure of the agent to that of the expert using the JS divergence, which endows some difficult optimization problems such as gradient vanishing. Our method uses the distance measurement of MMD-VKL, which provides better signals for the policy updating. Moreover, GAIL and VAIL adopted a sophisticated constraint on cost function specially designed for deriving a GAN-like objective. While the cost function constraint in our method is more natural, and the cost function class is more expressive in the sense that it spans over all possible RKHS spaces defined by all shift-invariant kernels.

GMMIL (Kim and Park 2018) matches the state-action distribution of an agent to that of an expert using MMD with a pre-defined Gaussian kernel. In our algorithm, we match the distribution using MMD with variational kernel learning, which could provide better signal for policy optimization.

Experiments

To demonstrate the performance of our algorithm, we first benchmark our algorithm against other state-of-the-art imitation learning algorithms over five high-dimensional control problems, integrated in OpenAI Gym (Brockman et al. 2016): Ant, HalfCheetah, Humanoid, HumanoidFlagrun, and Walker2D (Schulman et al. 2017). We then test our algorithm on a more complicated traffic control problem in a transportation system (Yang, Liu, and Dong 2019),

All the environments considered come with true reward (negative of cost) functions. We create the expert data through executing the trust region policy optimization (TRPO) (Schulman et al. 2015) algorithm over the true reward function, and then collect the expert trajectories through sampling actions from the expert policy and interacting with the environments. Detailed descriptions of the environments and the experimental setup, and additional experiments are given in the Appendix.

Continuous Control

For this set of experiments, we implement two instantiations of the procedure in Algorithm 1, a VAKLIL with a Gaussian kernel parameterization (VAKLIL-G), and a version using a random Fourier kernel (VAKLIL-F). We benchmark against four mainstream IL algorithms: apprenticeship learning (AL) (Abbeel and Ng 2004), generative adversarial IL (GAIL) (Ho and Ermon 2016), variational adversarial IL (VAIL) (Peng et al. 2018), and generative moment matching IL (GMMIL) (Kim and Park 2018). The policy networks in all algorithms are implemented with the same neural network structure.

The learning curves are plotted in Figure 2, and the statistics are summarized in Table 1. The performance is reported with the best performing parameters of each algorithm through grid searches. The empirical results show that our algorithm achieves the best performance in all scenarios, and converges faster than other algorithms. Our algorithms perform slightly better than other benchmarking algorithms in low dimensional control tasks such as HalfCheetah. However, in high dimensional control such as Walker2d and HumanoidFlagrun where the reward function is harder to learn, our algorithms perform much better. VAKLIL-F is slightly better than VAKLIL-G because VAKLIL-F enjoys more freedom in kernel learning, which leads to policy optimization over a wider reward function class. AL performs the worst because it has a strict assumption of the reward function classes. VAKLIL-F and VAKLIL-G outperform GAIL, VAIL, and

Metrics	**TRPE** ($\times 10^4$) ↑	**VOR** ↓	**VAH** ↑	**VAW** ↑	**VOT** ↑
Expert	90.68±5.32	5.21±3.48	49.86±20.03	64.77±4.15	57.3±0.47
VAKLIL-F	79.45±16.06	1.15±2.67	9.85±9.74	69.39±7.48	74.8±1.04
GAIL	56.70±2.69	0.49±1.17	59.92±12.10	37.76±4.64	31.7±0.85

Table 2: Comparisons in terms of total reward per episode (TRPE), number of vehicles on road per unit time (minutes) (VOR), number of vehicles at home after work hours per unit time (VAH), number of vehicles at work during work hours per unit time (VAW), and the number of vehicles arriving at work on time (VOT).

GMMIL because we use an MMD with variational kernel learning to measure the distance, which is more stable and can provide better signals to train the policy than using the JS distance or an MMD with a fixed kernel.

Traffic Control

The traffic control problem in a complex transportation system contains 101 state variables and 200 action variables. Through controlling the movement of vehicles, the goal is to let people driving in a more efficient manner, such as spending less time on the road, arriving at work on time, and spending more time at work during work hours.

We apply VAKLIL-F and GAIL to the traffic control task. The results are summarized in Table 2, which indicate that the learned policy of VAKLIL-F achieves higher scores in temrs of TRPE, VAW, and VOT than GAIL. By inspecting the scores, it is interesting to find that VAKLIL-F learns a policy that guides vehicles to work locations on time, and to stay at work during work hours; while GAIL learns a policy that suggests most of the vehicles staying at home.

Conclusion

In this paper, we developed VAKLIL, an imitation learning algorithm which optimizes the policy with more informative signals by defining the cost function in a learnable RKHS, and is more sample efficient and robust to overfitting. We benchmark VAKLIL against existing state-of-the-art algorithms on five OpenAI Gym environment and a complex transportation environment. Empirical results showed that our algorithm outperforms related algorithms in all scenarios.

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, 1. ACM.

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*.

Andrychowicz, M.; Wolski, F.; Ray, A.; Schneider, J.; Fong, R.; Welinder, P.; McGrew, B.; Tobin, J.; Abbeel, O. P.; and Zaremba, W. 2017. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, 5048–5058.

Arbel, M.; Sutherland, D.; Bińkowski, M.; and Gretton, A. 2018. On gradient regularizers for mmd gans. In *Advances in Neural Information Processing Systems*, 6700–6710.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

Bellemare, M. G.; Danihelka, I.; Dabney, W.; Mohamed, S.; Lakshminarayanan, B.; Hoyer, S.; and Munos, R. 2017. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*.

Berlinet, A., and Thomas-Agnan, C. 2011. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media.

Bińkowski, M.; Sutherland, D. J.; Arbel, M.; and Gretton, A. 2018. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. Openai gym. *arXiv preprint arXiv:1606.01540*.

Fukumizu, K.; Gretton, A.; Lanckriet, G. R.; Schölkopf, B.; and Sriperumbudur, B. K. 2009. Kernel choice and classifiability for rkhs embeddings of probability distributions. In *Advances in neural information processing systems*, 1750–1758.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A kernel two-sample test. *Journal of Machine Learning Research* 13(Mar):723–773.

Hadfield-Menell, D.; Milli, S.; Abbeel, P.; Russell, S. J.; and Dragan, A. 2017. Inverse reward design. In *Advances in neural information processing systems*, 6765–6774.

Ho, J., and Ermon, S. 2016. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, 4565–4573.

Jeon, W.; Seo, S.; and Kim, K.-E. 2018. A bayesian approach to generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, 7429–7439.

Kahn, G.; Villaflor, A.; Ding, B.; Abbeel, P.; and Levine, S. 2018. Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1–8. IEEE.

Kim, K.-E., and Park, H. S. 2018. Imitation learning via kernel mean embedding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Li, C.-L.; Chang, W.-C.; Cheng, Y.; Yang, Y.; and Póczos, B. 2017. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, 2203–2213.

Li, C.-L.; Chang, W.-C.; Mroueh, Y.; Yang, Y.; and Póczos, B. 2019. Implicit kernel learning. *arXiv preprint arXiv:1902.10214*.

Li, L.; Lv, Y.; and Wang, F.-Y. 2016. Traffic signal timing via deep reinforcement learning. *IEEE/CAA Journal of Automatica Sinica* 3(3):247–254.

Li, Y.; Song, J.; and Ermon, S. 2017. Infogail: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems*, 3812–3822.

Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937.

Peng, X. B.; Kanazawa, A.; Toyer, S.; Abbeel, P.; and Levine, S. 2018. Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow. *arXiv preprint arXiv:1810.00821*.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Rudin, W. 1962. *Fourier analysis on groups*, volume 121967. Wiley Online Library.

Sammut, C., and Webb, G. I. 2011. *Encyclopedia of machine learning*. Springer Science & Business Media.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *International Conference on Machine Learning*, 1889–1897.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Yang, F.; Liu, B.; and Dong, W. 2019. Optimal control of complex systems through variational inference with a discrete event decision process. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 296–304. International Foundation for Autonomous Agents and Multiagent Systems.

Yang, F.; Vereshchaka, A.; and Dong, W. 2018. Predicting and optimizing city-scale road traffic dynamics using trajectories of individual vehicles. In *2018 IEEE International Conference on Big Data (Big Data)*, 173–180. IEEE.