Generative Hybrid Representations for Activity Forecasting with No-Regret Learning

Jiaqi Guan*[1,2], Ye Yuan[1], Kris M. Kitani[1], and Nicholas Rhinehart*[1,3]

[1]Carnegie Mellon University [2]University of Illinois Urbana Champaign [3]UC Berkeley

{jiaqig, yyuan2, kkitani}@cs.cmu.edu, nrhinehart@berkeley.edu

Abstract

Automatically reasoning about future human behaviors is a difficult problem but has significant practical applications to assistive systems. Part of this difficulty stems from learning systems' inability to represent all kinds of behaviors. Some behaviors, such as motion, are best described with continuous representations, whereas others, such as picking up a cup, are best described with discrete representations. Furthermore, human behavior is generally not fixed: people can change their habits and routines. This suggests these systems must be able to learn and adapt continuously. In this work, we develop an efficient deep generative model to jointly forecast a person's future discrete actions and continuous motions. On a large-scale egocentric dataset, EPIC-KITCHENS, we observe our method generates high-quality and diverse samples while exhibiting better generalization than related generative models. Finally, we propose a variant to continually learn our model from streaming data, observe its practical effectiveness, and theoretically justify its learning efficiency.

1. Introduction

A key requirement for intelligent systems to safely interact with humans is the ability to predict plausible human behaviors. Additionally, they must be able to adapt to variability in behavior over time. However, forecasting a person's behaviors is generally difficult due to the immense set of possible behaviors that humans showcase. This makes it challenging to choose a unified representation for human behavior. Some behaviors are better modeled as continuous representations, for instance, a person's future trajectory. Other behaviors are more succinctly represented discretely, such as interacting with an object. Our goal is to develop an efficient predictive model for joint discrete-continuous spaces, which takes rich sensory information from egocentric videos as input to forecast a person's future behaviors.

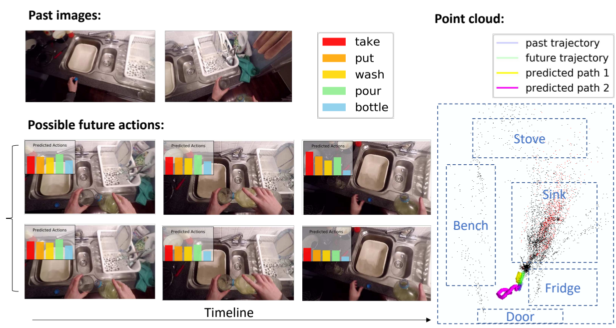

Figure 1. **Generative hybrid activity forecasting.** Our model generates possible future trajectories and actions with past trajectory and images as context. A point cloud is recovered with ORB-SLAM. Histograms show the possibilities of top 5 action classes.

For many applications based on a predictive model of future human behavior, it is important that the model is able to characterize the uncertainty of its predictions. A generative model can naturally represent uncertainty and is also well-suited for modeling a hybrid representation of human behavior. Thus, we propose a generative model that can represent the joint distribution of discrete and continuous behaviors by leveraging recent success in generative modeling. Unlike some popular generative models (*e.g.* GANs [12] and variational autoencoders [21]), our method can compute exact likelihoods, which makes it possible to precisely evaluate the model's predictions of future behaviors. It is part of a family of methods known as *invertible generative models* [5, 13, 20]. We learn a generative model of discrete actions by applying the Gumbel-Softmax trick [29], and condition this model on continuous samples produced by an invertible generative trajectory model [36]. We show how we can jointly learn both models efficiently. The results on a large-scale egocentric dataset, EPIC-KITCHENS [4], demonstrate the advantage of our model in joint trajectory-

*Work done primarily while Jiaqi Guan and Nicholas Rhinehart were at CMU. Jiaqi Guan's current email: jiaqi@illinois.edu action forecasting over other generative models and discriminative models. To enable our model to learn optimally from streaming data we employ online learning theories [43]. In particular, we apply a modified objective to finetune a subset of the model's parameters using a no-regret online learning algorithm. We prove our method's effectiveness theoretically, and observe its online performance matches these theoretical expectations. Example predictions of our method are shown in 1.

We present the following contributions:

1. **Generative hybrid representations:** We propose a generative approach to egocentric forecasting that jointly models trajectory and action distributions. Our experiments on the EPIC-KITCHENS dataset show that our method outperforms both discriminative and generative baselines.
2. **Exact learning and evaluation:** Our model can compute the probability density function (PDF) exactly and also enables optimization of model sample-based metrics (*e.g.,* reverse cross entropy), which renders learning and inference of people's future trajectory and action more efficient.
3. **Theoretically justified no-regret online fine-tuning:** We extend our model to learn online with a simple, yet effective fine-tuning process. We demonstrate that it is theoretically efficient, which enables the model to learn from data that arrives continuously and the average *regret* will approach to zero with time elapsing.

2. Related Work

We propose a generative model to jointly forecast future trajectories and actions under the first-person vision setting. We begin by discussing work related to our data domain, task, and model.

First-person vision: As wearable cameras become more accessible in our daily lives, a growing body of work is using them for understanding human behaviors [7, 26, 42, 27, 32, 51]. The rich visual information encoded in first-person videos can also be used to predict the subject's attention [27, 53] and their interactions with the environment.

Trajectory Forecasting: *Third-person* trajectory forecasting has enjoyed significant research attention recently. The approach in [25] predicts future trajectories of wide-receivers from surveillance video. A large body of work has also used surveillance video to predict future pedestrian trajectories [49, 28, 2, 22]. Deterministic trajectory modeling has been used for vehicle [17] and pedestrian [1, 37, 50] trajectory prediction. Due to the uncertain nature of future trajectories, modeling stochasticity can help explain multiple plausible trajectories with the same initial context. Several approaches have tried to forecast distributions over trajectories [24, 9]. [36] proposed a generative approach to model vehicle trajectories. A relative small amount of work has investigated trajectory forecasting from *first-person* videos. [44] predicts the future trajectories of the camera wearer by constructing an EgoRetinal map.

These approaches employed continuous representations in the batch learning setting, while our model uses both discrete and continuous representations in both the batch and online learning settings.

Action Forecasting: Classification-based approaches [16, 23, 41, 40] are popular in action forecasting. Many activities are best represented as categories. [10] proposed an encoder-decoder LSTM model to predict future actions. Other work has also tried to forecast more generalized action such as gaze [55], user-object interactions [8] and the position of hands and objects [6]. In [35], online inverse reinforcement learning (IRL) is used to model a person's goals and future trajectories. IRL has also been applied to forecast the behaviors of robots [33], taxis [57], and pedestrians [22]. Some work has investigated non-discriminative modeling of future actions. [45] devised a deep multi-modal regressor to allow multiple future predictions. [6] uses a variational autoencoder (VAE) to model the distribution of possible future actions. Whereas prior activity forecasting approaches reason about actions only, our method reasons jointly about actions and trajectories.

Generative Models: Deep generative models, *e.g.* [12, 21], are a powerful unsupervised modeling approach. To enable efficient learning of deep generative models of categorical distributions, [29] proposed the Gumbel-Softmax trick to backpropagate gradients through these distributions. There has been work that uses generative models to address the uncertainty in both trajectory [24, 9, 36, 52, 48, 54] and action forecasting [6, 45]. Unlike the prior approaches, our method jointly generates the future trajectories and actions.

Online Learning: The field of online learning studies how to learn effectively from streaming data [43], but these approaches are rarely used in computer vision problems. In [35], online inverse reinforcement learning is performed with visual data. In contrast, our approach is based on imitation learning without reward modeling. In [39, 38], interactive imitation learning is framed as a online learning problem. Our approach, while a form of imitation learning, is not interactive. It observes expert behavior (human behaviors) and makes predictions that the human does not interact with.

3. Generative Hybrid Activity Forecasting

3.1. Problem Formulation

Our goal is to model the true joint distribution $p(x, a|\phi)$ of a person's future trajectory $x \in \mathbb{R}^{T\times 3}$ in 3D and actions $a \in \{0, 1\}^{T\times C_a \times 2}$ from egocentric videos with a learned joint distribution $q(x, a|\phi)$, where ϕ is the context informa-

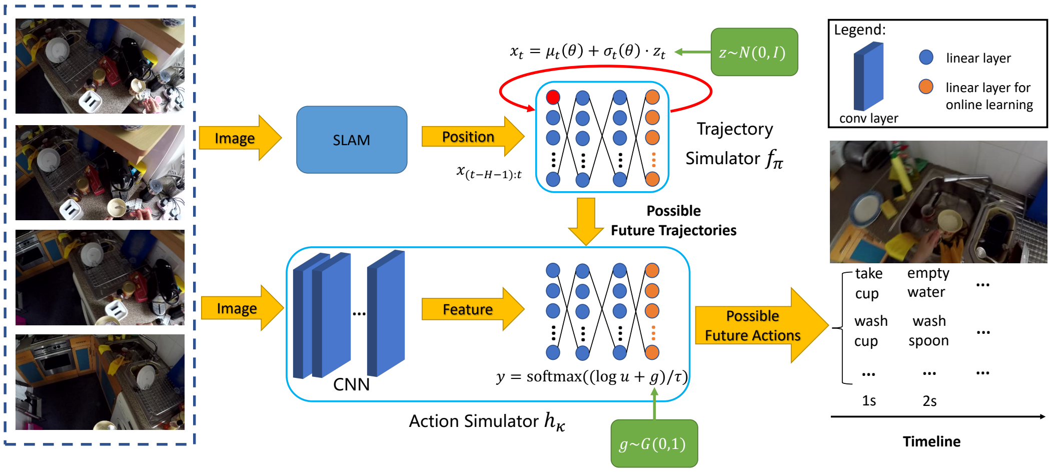

Figure 2. **Our proposed model.** ORB-SLAM [30] is used to extract positions from videos. Trajectory simulator f_π takes past positions and noise sequence from Gaussian distribution to generate future trajectory. Action simulator h_κ takes past images and positions as well noise sequence from Gumbel distribution to produce future actions.

tion, T is the forecasting horizon, and C_a is the number of action classes (with each class modeled with 2 values using a one-hot encoding). The context information ϕ includes past egocentric video frames $V_{-P:0}$ and positions $x_{-P:0}$, where P is the observation horizon.

As x and a use different representations (continuous vs. discrete), we further factorize the joint distribution by conditioning a on x i.e. $q(x, a|\phi) = q(x|\phi)q(a|x, \phi)$. Learning this model of future behavior via divergence minimization is akin to *imitation learning* [11, 19]. We use *one-step* policies π for generating trajectory x and κ for generating actions a, and samples from $q_\pi(x|\phi)$ and $q_\kappa(a|x, \phi)$ can be obtained by repeatedly sampling T times from π and κ. These policies parameterize each generative model. Our training data is a set of episodes denoted $\{(x, a, \phi)_n\}_{n=1}^N$, which are samples from the (unknown) data distribution of the person's behavior $p(x, a|\phi)$. We use this data to train the policies π and κ, thereby learning $q(x, a|\phi)$.

3.2. Complementary Cross Entropy Loss

A desired feature of forecasting models is to generate both *diverse* and *precise* predictions. Following [36], we construct a complementary cross-entropy loss to train our trajectory-action distribution $q(x, a|\phi)$:

$$\mathcal{L} = \underbrace{\mathbb{E}_{(x,a)\sim p} - \log q(x, a|\phi)}_{H(p,q)} + \beta \underbrace{\mathbb{E}_{(x,a)\sim q} - \log \tilde{p}(x, a|\phi)}_{H(q,\tilde{p})}, \tag{1}$$

where $\tilde{p}$ is an approximation to the data distribution p, which we will discuss it in detail in Sec 3.6. β is a weighting factor. The forward cross entropy term $H(p, q)$ encourages the distribution q to cover all modes of p and thus increases sample diversity. The reverse cross entropy term $H(q, \tilde{p})$ penalizes samples far from the data distribution $\tilde{p}$ to improve sample quality. The joint use of them promotes both diversity and quality of samples. We use β to control the trade-off between diversity and precision.

With the factorization $q(x, a|\phi) = q_\pi(x|\phi)q_\kappa(a|x, \phi)$, the forward and reverse cross entropies can be rewritten as

$$\begin{aligned} H(p,q) &= \underbrace{-\mathbb{E}_{x\sim p} \log q_\pi(x|\phi)}_{H(p,q_\pi)} \underbrace{-\mathbb{E}_{(x,a)\sim p} \log q_\kappa(a|x,\phi)}_{H(p,q_\kappa)}, \\ H(q,\tilde{p}) &= \underbrace{-\mathbb{E}_{x\sim q_\pi} \log \tilde{p}(x|\phi)}_{H(q_\pi,\tilde{p})} \underbrace{-\mathbb{E}_{x\sim q_\pi, a\sim q_\kappa} \log \tilde{p}(a|x,\phi)}_{H(q_\kappa,\tilde{p})}. \end{aligned} \tag{2}$$

This decomposition disentangles the cross entropies for trajectory and actions, allowing us to learn the policy π and κ separately. The optimization of $H(p, q)$ requires us to compute q and the optimization of $H(q, \tilde{p})$ requires us to sample from q. Different from GANs [12] (likelihood-free learning) and VAEs [21] (optimize the evidence lower bound), we propose an *invertible generative model*, which enables us to both compute the likelihood of $q(x, a|\phi)$ exactly and generate samples from $q(x, a|\phi)$. The model details will be illustrated in Sec 3.3, 3.4 and 3.5.

3.3. Trajectory Cross Entropy

We employ an invertible trajectory generative model by constructing a differentiable, invertible function $f_\pi(z; \phi)$: $\mathbb{R}^{T\times 3} \to \mathbb{R}^{T\times 3}$. This function maps a noise sequence $z = [z_1, \ldots, z_T]$ from a Gaussian distribution $\mathcal{N}(0, I_{3\times 3})$ and the scene context ϕ to a trajectory $x = [x_1, \ldots, x_T]$. f_π is implemented by a θ-parametrized per-step policy π. At each time step t, π takes in a per-step context ψ_t, containing past positions $x_{t-P:t-1}$, and outputs the mean μ_t and an invertible covariance matrix σ_t, and simulate the current position x_t with noise z_t: $x_t \triangleq \mu_t(\psi_t; \theta) + \sigma_t(\psi_t; \theta)\, z_t$. Since σ_t is invertible, π defines a bijection between z_t and

x_t, and f_π defines a bijection between x and z.

q_π then follows from the change-of-variables formula for multivariate integration [34, 5, 13, 20]:

$$q_\pi(x|\phi) = \mathcal{N}\left(f_\pi^{-1}(x;\phi)\right) \left|\det J_{f_\pi}\left(f_\pi^{-1}(x;\phi)\right)\right|^{-1}, \quad (3)$$

where $J_{f_\pi}(f_\pi^{-1}(x;\phi))$ is the Jacobian of f_π evaluated at $f_\pi^{-1}(x;\phi)$. Thus, the forward cross entropy can be rewritten as

$$H(p, q_\pi) = -\mathbb{E}_{x\sim p} \log \frac{\mathcal{N}\left(f_\pi^{-1}(x;\phi)\right)}{\left|\det J_{f_\pi}\left(f_\pi^{-1}(x;\phi)\right)\right|}. \quad (4)$$

The reparameterization also greatly simplifies the differentiation of $H(q_\pi, \tilde{p})$ w.r.t. policy π. Instead of sampling from q_π, we can sample from $\mathcal{N}$ and rewrite the reverse cross entropy as Eq.(5). z is the source of uncertainty for generating diverse samples.

$$H(q_\pi, \tilde{p}) = -\mathbb{E}_{z\sim\mathcal{N}} \log \tilde{p}\left(f_\pi(z;\phi)|\phi\right). \quad (5)$$

3.4. Action Cross Entropy

For the action forecasting, at each step t each single action class c is represented as $a_{t,c} \in \{0,1\}^2$ which is a one-hot vector indicating whether this action happens ($[0,1]$) or not ($[1,0]$). Since actions are discrete variables, we use Gumbel-Softmax distributions [18] to reparameterize actions. We build a simulator $h_\kappa(g;\phi) : \mathbb{R}^{T\times C_a\times 2} \rightarrow \{0,1\}^{T\times C_a\times 2}$, which maps noise sequences g sampled from Gumbel distribution $\mathcal{G}(0,1)$ to actions a. The noise sequence g, as a key part of the Gumbel-Softmax reparameterization – a continuous, differentiable approximation to Gumbel-Max, provides an efficient way to draw samples from a categorical distribution.

The per-step action forecasting context χ_t consists of past images $V_{-P:0}$ and past positions $x_{t-P:t-1}$. The per-step policy κ outputs action probabilities u_t with χ_t, and simulate the current action a_t with noise g_t:

$$a_{t,c,i} \triangleq \frac{\exp\left((\log(u_{t,c,i}(\chi_t;\theta)) + g_{t,c,i})/\tau\right)}{\sum_{j=1}^{2}\exp\left((\log(u_{t,c,j}(\chi_t;\theta)) + g_{t,c,j})/\tau\right)},$$

where $i \in \{1,2\}$, $c \in \{1,\ldots,C_a\}$, and $t \in \{1,\ldots,T\}$, τ is the temperature of Gumbel-Softmax distribution.

According to the probability density function of the Gumbel-Softmax distribution [18], the action forward cross entropy can be rewritten as

$$H(p, q_\kappa) = - \mathbb{E}_{(x,a)\sim p} \sum_{t,c} \log \tau \left(\sum_{i=1}^{2} \frac{u_{t,c,i}(\chi_t)}{a_{t,c,i}^{\tau}}\right)^{-2} \prod_{i=1}^{2}\left(\frac{u_{t,c,i}(\chi_t)}{a_{t,c,i}^{\tau+1}}\right), \quad (6)$$

For the reverse cross entropy, using Gumbel-Softmax reparameterization, it can be rewritten as

$$H(q_\kappa, \tilde{p}) = -\mathbb{E}_{g\sim\mathcal{G}} \sum_{t,c,i} \log \tilde{p}(a_{t,c,i}|x,\phi). \quad (7)$$

The overall procedure of training the batch model is shown in Algorithm 1.

Algorithm 1 Offline Generative Hybrid Activity Forecasting

```
Require: Training dataset {(x, a, φ)_n}_{n=1}^N; Batch size B;
    Trajectory simulator f_π; Action simulator h_κ
Randomly initialize f_π and h_κ with parameter θ
repeat
    for each mini-batch examples (x, a, φ)_{i:i+B} do
        Calculate H(p, q_π) with Eq. (4) (6)
        Sample z ~ N; Generate trajectory x̂ = f_π(z; φ)
        Calculate H(q_π, p̃) with Eq. (5)
        Sample g ~ G; Generate actions â = h_κ(g; φ)
        Calculate H(q_κ, p̃) with Eq. (7)
        Update θ by optimizing Eq.(1)
    end for
until θ converge
return θ as θ̂
```

Algorithm 2 Online Generative Hybrid Activity Forecasting

```
Require: Trajectory simulator f_π; Action simulator h_κ;
    Pre-trained weights θ̂
Initialize f_π, h_κ with θ̂
Fix all parameters except the linear layer θ_0 at the end
for each new example do
    [x, y, z] ← slam.track()
    Calculate H(p, q) with Eq. (4) (6)
    Sample z ~ N; Generate trajectories x̂ = f_π(z; φ)
    Calculate H(q, p̃) with Eq. (8)
    Finetune θ_0 by optimizing Eq. (9) with SGD
end for
```

3.5. Policy Modeling

Trajectory Modeling. For the trajectory policy π, we use a recurrent neural network (RNN) with gated recurrent units [3] that maps context ψ_t to $\hat{\mu}_t$ and S_t. We use the matrix exponential [31] to ensure the positive definiteness of σ_t: $\sigma_t = \operatorname{expm}\left(S_t + S_t^T\right)$. The network architecture is shown in Figure 2. We provide more architectural details in the supplementary material.

Action Modeling. Our action policy κ maps context χ_t to action probabilities u_t, and is based on the idea of Temporal Segment Networks [47] with a ResNet-50 [15] backbone. The past images $V_{-P:0}$ we observe are divided into K segments and an image is selected randomly from each segment. These images are passed through a ResNet independently to get the class scores. Another fully-connected layer is built on top of the ResNet to fuse these class scores to yield segmental consensus, which serves as a useful feature in our action forecasting. In the meanwhile, the past trajectory $x_{t-P:t-1}$ also includes useful information about what kind of actions people may perform. Thus, we add a MLP which takes the segmental consensus and the past trajectory as inputs to generate the action probabilities u_t.

3.6. Prior Distribution Approximation

It is challenging to evaluate $H(q_\pi, p)$ without the PDF of p (here, the density function of future behavior). We propose a simple approach to estimate it using the training data. For trajectory $H(q_\pi, p)$, we build $\tilde{p}$ as a sequence of unimodal normal distributions with ground-truth trajectory $\tilde{x}$ as means, *i.e.*, $\tilde{p}(x|\phi) = N(\cdot|\tilde{x}; \sigma I)$. In fact, this is identical to adding a mean squared distance penalty between the predicted trajectories and expert trajectories. For action $H(q_\kappa, p)$, we first assume that if an action occurs at time t, then the same action has a higher probability happening at time steps closer to t. Based on this assumption, we can also view each action happening at t as a unimodal normal distribution in the time dimension. If the action spans several time steps, we take the max of the distributions induces by different t. As a result, we obtain the approximate action prior distribution $\tilde{p}(a|x, \phi)$. Note that this action prior does not actually depend on the trajectory x, this is partly due to the difficulty of defining a conditioned prior distribution. On the other hand, our reverse cross entropy can be seen as a regularization of trajectories and action, and the independent version can achieve this.

3.7. Online No-regret Learning

To apply the proposed framework to an online scenario where the policies are learned over time, we would like to ensure that the learning process is guaranteed to converge to the performance of the strongest model. We can evaluate the relative convergence properties of an online learning algorithm through regret analysis. To leverage known proofs of no-regret learning, one should ensure that the model and loss function being used is convex. To this end, we pretrain the network and fix parameters of nonlinear layers. We slightly adjust the trajectory reverse cross entropy as Eq. (8) and perform online gradient descent on the loss function in Eq. (9) by fine-tuning the parameters of the last linear layer. The regret is computed with respect to a model family, and the model family we consider is one of pre-trained representations that are fine-tuned to adapt to online performance. The detailed online learning parameterization is explained in supplementary material.

$$H\left(q_\pi, \tilde{p}\right)^{\text{adj}} = -\mathbb{E}_{x_{1:t-1} \sim p, x_{t:T} \sim q_\pi} \log \tilde{p}\left(x|\phi\right) , \qquad (8)$$

$$L_{\text{online}} = H\left(p, q_\pi\right) + H\left(q_\pi, \tilde{p}\right)^{\text{adj}} + H\left(p, q_\pi\right) . \qquad (9)$$

In general, the regret R_T of an online algorithm is defined as: $R_T = \sum_{t=1}^{T} l_t\left(\xi_t; \theta_t\right) - \min_{\theta^*} \sum_{t=1}^{T} l_t\left(\xi_t; \theta^*\right)$, where ξ_t and l_t is the input and the loss at time step t separately. We can prove our forward cross entropy loss is convex with respect to the parameters of the finetuned linear layer. If we further constrain the parameter's norm $\|\theta\|_2 \leq B$ and the gradient's norm $\|\nabla_\theta\|_2 \leq L$, then the regret of our online algorithm is bounded [43] as: $R_T \leq BL\sqrt{2T}$.

Since the bound is sub-linear in T, the average regret R_T/T approaches zero as T grows, so it is a no-regret algorithm. The overall online learning procedure is shown in Algorithm 2. The detailed proof of the no-regret property is given in the supplementary material and the empirical results are shown in the experiments.

4. Experiments

We evaluate our models and baselines on the EPIC-KITCHEN [4] dataset. In this section, we first describe the dataset and related data processing steps. We then introduce the baselines that we use to compare our model with and the metrics to evaluate the performance of trajectory forecasting and action forecasting. In the experiments, we perform both batch and online experiments with the goal to validate the following hypotheses: (1) Since the trajectory-action joint model make actions conditioned on positions, the extra position information should help achieve better action forecasting performance than separately trained model. (2) The reverse cross entropy terms for trajectory and actions in our loss function should help improve sample quality. (3) The ability of evaluating the exact PDF of the trajectory and action distribution should help our model achieve lower cross entropies and higher sample quality than other generative methods that do not optimize the exact PDF such as CVAE. (4) The generative model should have the ability to generate samples with higher quality than discriminative models since it considers the multi-modal nature of future behavior and can generate multiple reasonable samples during the evaluation, while discriminative models can not. (5) We want to show from an empirical perspective that our online learning method is effective and no-regret.

4.1. Data Description

We evaluate our method on the EPIC-KITCHENS dataset [4]. First, we use ORB-SLAM [30] to extract the person's 3D positions from the egocentric videos. For each video, we start to collect positions when the pose-graph is stable and no global bundle adjustment is performed. We also scale positions with the first 30-second results by assuming that the person's activity range in each video is similar to alleviate the scale ambiguity caused by the initialization of ORB-SLAM. Then, we extract examples with successive 7-second interval. Those discontinuous examples (such as when tracking gets lost) are dropped out. In each 7-second example, we use the past 2 seconds as context to predict the future trajectory and actions in the next 5 seconds. We down-sample original data to 5 fps for position, 2 fps for images, and 1 fps for actions. Thus, the context we use to train the model contains 10 past positions and 4 past images. We filter actions to guarantee that each action occurs at least 50 times and drop videos which includes less than 5 examples. Finally, we use 4455 examples in total, which come from 135 videos. The number of action classes is 122 with 39 verbs and 83 nouns. Since the annotations of the test set are not available, we randomly split the original training videos to training, validation, and test with the proportion of 0.7, 0.1, 0.2. At the same time, we ensure each action occurs in both training set and test set and the examples in different sets come from different videos.

We predict verbs and nouns separately instead of predicting the pairs of them, which is different from the setting in [4]. This is because first the combination of verbs and nouns would create too many action classes and each class would have few samples; second, there are often multiple actions taking place at the same time in the dataset, which leads to our multi-label classification formulation.

4.2. Baselines and Metrics

Baselines The baselines we use include two generative models and a discriminative model:

- **Direct Cross Entropy (DCE):** a generative model that uses a sequence of Gaussian to model the trajectory distribution, and a sequence of Bernoulli distributions conditioned on the trajectory to model the action distribution.
- **Conditional Variational Autoencoder (CVAE):** an auto-regressive variant VAE-based generative model. We use the Gumbel-Softmax to model the action distribution.
- **Mixed Regression and Multi-label Classification (MRMC):** a discrimintative model trained by minimizing the mean squared error of trajectories and the binary cross entropy of actions.

For all baseline models, we follow the same network structure as our model to process past positions and images context. Detailed info can be found in the supplementary.

Metrics We use the following metrics to comprehensively evaluate our method and other baselines:

- **Forward Cross entropy:** for trajectory and action forecasting, we use their corresponding forward cross entropies $H(p, q_\pi)$ and $H(p, q_\kappa)$ to evaluate how well the policy mimics the behaviors of the expert.
- **minMSD and meanMSD:** for trajectory forecasting, we also include two common sample-based metrics often used in generative models – minMSD and meanMSD [24, 46, 14, 36]. minMSD computes the smallest distance from K samples to the ground-truth x: $\min_k \|\hat{x}^k - x\|^2$. Thus, minMSD evaluates the quality of the best sample. In contrast, meanMSD evaluates the overall quality of all K samples via $\frac{1}{K}\sum_{k=1}^{K} \|\hat{x}_k - x\|^2$. The combined use of these two metrics evaluates the quality of generated trajectories comprehensively. We sample 12 trajectories for each example. For discriminative models, we directly report the regression results as minMSD and meanMSD.
- **Precision, Recall and F-1 score:** for action forecasting, since the action space is large and we need to forecast actions in 5 seconds per example, the exact matching accuracy is not be a good metric. Instead, we calculate the example-based precision and recall as [56]. One special case is that if there is no ground-truth action or predicted action happening at some time step, the denominator will be zero. If this happens, the precision and recall is 1 only if $tp = fp = fn = 0$, where tp, fp, fn is the number of true positives, false positives, and false negatives, otherwise the precision and recall is 0. To consider both precision and recall, we also calculate F-1 score as $F_1 = \frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$. As action distribution is conditioned on the forecasted trajectory, we first sample 12 trajectories, and for each trajectory we sample the action (for each action class, the action happens if its logit is greater than 0.5) and average the metrics across the trajectories. For discriminative models, we directly report the multi-label classification results.

4.3. Batch Forecasting Results

Our full model is a joint forecasting model which makes actions conditioned on the trajectory, and it is trained using the complementary loss function in Eq. (1). To test whether the joint modeling of trajectory and action distribution help improve forecasting performance, we also train a trajectory forecasting model and an action forecasting model separately. We also evaluate a variant of our method by using only the forward cross entropy for both action and trajectory. The results are summarized in Table 1.

First, we can see that our joint forecasting model (g) outperforms separately trained models (e) in action forecasting metrics (cross entropy, precision, recall, and F1-score), so our factorization – conditioning actions on the trajectory indeed helps. Hypothesis (1) is supported. Comparing (e)(g) to (f)(d), we can see the quality of both trajectory samples and actions samples are better after using the reverse cross entropy, which justifies its use in the loss function and also demonstrates the effectiveness of our designed prior data distribution. Hypothesis (2) is supported. Furthermore, our methods outperforms other generative baselines (b)(c) in terms of most metrics, especially forward cross entropy. This is due to the fact that our method has more modeling power than DCE, and can evaluate the exact PDF of trajectory and action distribution instead of optimizing the variational lowerbound like CVAE does. Our model does not outperform *MRMC* in the meanMSD metric and *DCE* in the Recall metric, but we note that: 1. The *MRMC* model can not conduct sampling, so it leads to lower meanMSD than all of other generative models; 2. The *DCE* model actually cannot generate good enough examples, which is indicated by the low precision and low F1 score, even if it has high recall; 3. All baselines make actions condi-

Method	Trajectory Forecasting			Action Forecasting			
	$H(p, q_\pi)$ (↓)	minMSD(↓)	meanMSD(↓)	$H(p, q_\kappa)$ (↓)	Precision(↑)	Recall(↑)	F_1 (↑)
(a) MRMC	-	0.392	**0.392**	-	40.64	32.12	35.88
(b) DCE	-26.93	0.539 ± 0.010	1.870 ± 0.094	-40.22	11.04 ± 3.11	$\mathbf{39.31} \pm 2.10$	17.24 ± 2.49
(c) CVAE	$\leq$ -129.78	0.319 ± 0.008	1.394 ± 0.085	$\leq$ -135.21	38.48 ± 0.09	31.03 ± 0.04	34.38 ± 0.06
(d) Ours (S)-F	-288.26	0.304 ± 0.017	1.553 ± 0.077	-192.48	39.06	29.97	33.92
(e) Ours (S)	-275.81	$\mathbf{0.286} \pm 0.007$	0.915 ± 0.088	-192.31	39.97	29.80	34.14
(f) Ours (J)-F	**-298.92**	0.291 ± 0.017	1.446 ± 0.087	-192.53	42.89 ± 0.33	32.50 ± 0.29	36.98 ± 0.30
(g) Ours (J)	-298.47	0.293 ± 0.004	0.971 ± 0.078	**-192.57**	$\mathbf{44.10} \pm 0.11$	33.39 ± 0.07	$\mathbf{37.90} \pm 0.09$

Table 1. **Batch results on the EPIC-KITCHENS dataset.** For sample-based metrics, mean $\pm$ std is reported. MRMC: Mix Regression and Multi-label Classification (discriminative model). DCE: Direct Cross Entropy (generative model). CVAE: Conditional Variational Autoenconder (generative model). For our model, S denotes separate training of trajectory policy and action policy. J denotes joint training. F denotes the model is trained with forward cross entropy only. (↓)/(↑) denotes a metric for which lower/higher scores are better.

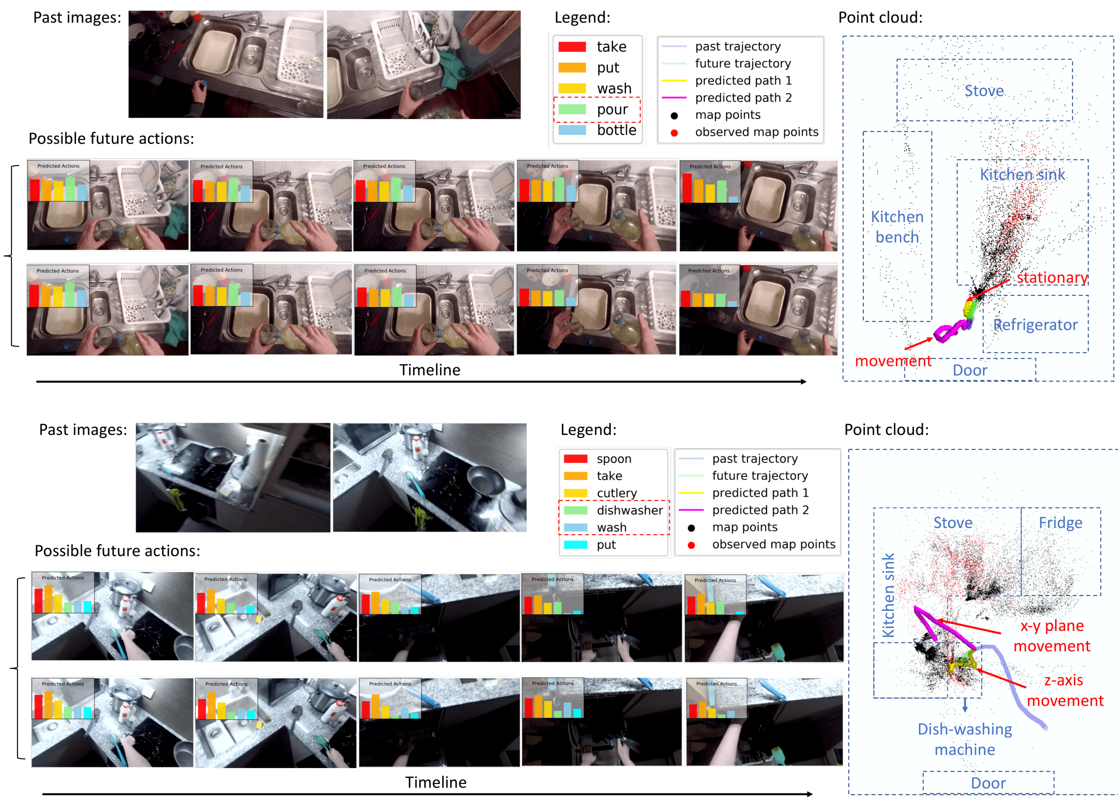

Figure 3. **Forecasting results visualization.** Visualization of two examples. It shows how the forecasted trajectory influences the action distribution. In each example, the left top shows observed images , the left bottom shows action distributions corresponding to two forecasted sample trajectories, and the right shows the point cloud of the scene and the forecasted trajectories (Red/Black points: Observed/Unobserved map points).

tioned on positions, so it is fair to compare *Ours(J)* with baselines, which shows better performance except for two aforementioned special cases. Hypothesis (3) is supported. Finally, our method also performs better than the discriminative baseline *MRMC*, because it fails to model the multi-modal nature of the future behavior. Fig. 4 illustrate this point further. We can see that our model continuously outperforms the discriminative model in terms of recall when we force the model output actions with top K (K is from 1 to 10) probabilities. The visualization example shows an environment with uncertainty. Given past information, we are actually not sure which actions (*wash hand*, *close tap*, *take cloth* or *dry hand*) will happen. Our model assigns relatively high confidence on these probable future actions but the discriminative model only focuses on two actions – *wash* and *cloth*. Thus, hypothesis (4) is also supported.

Fig. 3 shows visualization results of two examples. For each example, we show two sampled trajectories and their corresponding action distribution. In all these two examples, the forecasted trajectory influences the action distribu-

Experiment	Method	Trajectory Forecasting			Action Forecasting			
		$H(p, q_\pi)$ (↓)	minMSD(↓)	meanMSD(↓)	$H(p, q_\kappa)$ (↓)	Precision (↑)	Recall (↑)	F_1 (↑)
(i) Train→Test	Pre-online	-298.47	0.293 ± 0.004	0.971 ± 0.078	-192.57	44.10 ± 0.11	**33.39** ± 0.07	37.90 ± 0.09
	Online	**-299.66**	**0.283** ± 0.004	**0.963** ± 0.063	**-192.59**	**45.27** ± 0.10	32.90 ± 0.07	**38.11** ± 0.10
(ii) Test→Train	Pre-online	-204.23	0.280 ± 0.005	0.560 ± 0.080	-181.80	20.70 ± 0.03	20.28 ± 0.02	20.49 ± 0.02
	Online	**-220.38**	**0.230** ± 0.004	**0.497** ± 0.091	**-184.89**	**22.76** ± 0.05	**22.05** ± 0.04	**22.40** ± 0.05

Table 2. **Online learning results.** *Pre-online* denotes the results on the streaming data before online learning. *Online* denotes the results on the streaming data across online learning. Experiment $A \rightarrow B$ means we pretrain the model on the data set A and perform online learning on the data set B. (↓)/(↑) denotes a metric for which lower/higher scores are better.

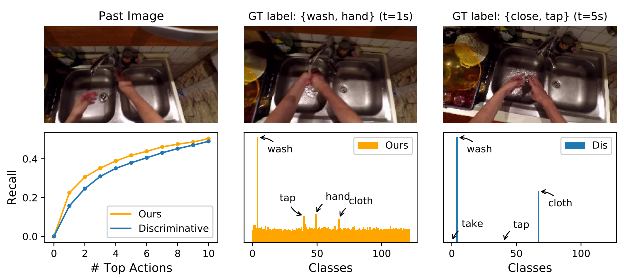

Figure 4. **Top-K visualization.** The first row is an example of uncertain future behavior. The bottom left plot shows the recall value of our model and the discriminative model if we force the model to output actions with top K probabilities (K is from 1 to 10). The other two plots in the bottom row show the action class probabilities of our model and the discriminative model separately. Our model shows better performance in handling uncertainty.

tion in a meaningful way. In the first example, the person is going to pour the drink. We can see that the person moves less in the first forecasted trajectory than the second one. As a result, the first trajectory has a higher probability for pouring because people tend to stay still when they are pouring something. In the second example, the person is going to take the cutlery and insert them into the dishwasher. In the first predicted trajectory, the person's position changes a lot along the z-axis, and the model predicts that the person is more likely to put the cutlery into the dishwasher. In contrast, the positions change a lot in the ground plane (xy-plane) in the second forecasted trajectory, and the model predicts that the person is more likely to wash something as it requires more horizontal movements.

4.4. Online Forecasting Results

We conduct two online learning experiments to verify the effectiveness of our model to learn from streaming data. We pretrain the model on the training set and perform online learning on the test set in (i), and inversely in (ii). In both experiments, we only finetune additional linear layers during online learning. Pre-online learning and online learning results are shown in Table 2. It can be seen that in both experiments, the model obtained after online learning outperforms the original model which shows the effectiveness of our online learning algorithm. Additionally, comparing (ii) with (i), we can also see that with more data observed, the relative improvement from online learning will be more significant. We also analyze the regret of our model. We train the online models and corresponding hindsight models using Eq. (9). The average regret curve of the forward experiment is shown in Fig. 5. We can see that the average regret curve converges to zero as more examples are observed, which proves that our model is no-regret. Hypothesis (5) is also supported. The theoretical analysis of no-regret can be found in the supplementary.

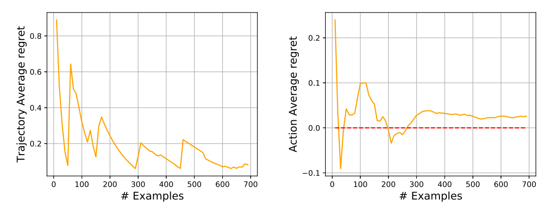

Figure 5. **Average regret curve.** We compare our online learning model with the hindsight model to calculate the average regret. The average regret of both trajectory forecasting (left) and action forecasting (right) show convergence towards zero, which supports the claim that our online learning method is no-regret empirically.

5. Conclusion

We proposed a novel generative model to represent hybrid continuous and discrete state for first-person activity forecasting. We model discrete actions conditioned on continuous trajectories and learn a deep generative model by minimizing a symmetric cross entropy loss. Our model can generate both precise and diverse future trajectories and actions based on observed past images and positions. The results on EPIC-KITCHENS dataset shows our method outperforms related generative models and discriminative models. Our model can also be easily adapted to no-regret online learning, which creates more application possibilities in complex real-world scenarios. A possible future work is the united representation of continuous and discrete variables with the help of discrete normalizing flow models, instead of factorizing the joint distribution to make actions conditioned on trajectories.

Acknowledgement. This work was sponsored in part by Center for Machine Learning and Health PhD Fellowship, and JST CREST (JPMJCR14E1).

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016. 2

[2] Lamberto Ballan, Francesco Castaldo, Alexandre Alahi, Francesco Palmieri, and Silvio Savarese. Knowledge transfer for scene-specific motion prediction. In *European Conference on Computer Vision*, pages 697–713. Springer, 2016. 2

[3] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 4

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. *arXiv preprint arXiv:1804.02748*, 2018. 1, 5, 6

[5] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. *arXiv preprint arXiv:1605.08803*, 2016. 1, 4

[6] Chenyou Fan, Jangwon Lee, and Michael S Ryoo. Forecasting hand and object locations in future frames. *CoRR, abs/1705.07328*, 2017. 2

[7] Alireza Fathi, Ali Farhadi, and James M Rehg. Understanding egocentric activities. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 407–414. IEEE, 2011. 2

[8] Antonino Furnari, Sebastiano Battiato, Kristen Grauman, and Giovanni Maria Farinella. Next-active-object prediction from egocentric videos. *Journal of Visual Communication and Image Representation*, 49:401–411, 2017. 2

[9] Enric Galceran, Alexander G Cunningham, Ryan M Eustice, and Edwin Olson. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction. In *Robotics: Science and Systems*, volume 1, 2015. 2

[10] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Red: Reinforced encoder-decoder networks for action anticipation. *arXiv preprint arXiv:1707.04818*, 2017. 2

[11] Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. *arXiv preprint arXiv:1911.02256*, 2019. 3

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1, 2, 3

[13] Will Grathwohl, Ricky TQ Chen, Jesse Betterncourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018. 1, 4

[14] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018. 6

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[16] Minh Hoai and Fernando De la Torre. Max-margin early event detectors. *International Journal of Computer Vision*, 107(2):191–202, 2014. 2

[17] Ashesh Jain, Avi Singh, Hema S Koppula, Shane Soh, and Ashutosh Saxena. Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 3118–3125. IEEE, 2016. 2

[18] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 4

[19] Liyiming Ke, Matt Barnes, Wen Sun, Gilwoo Lee, Sanjiban Choudhury, and Siddhartha Srinivasa. Imitation learning as f-divergence minimization. *arXiv preprint arXiv:1905.12888*, 2019. 3

[20] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10236–10245, 2018. 1, 4

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1, 2, 3

[22] Kris M Kitani, Brian D Ziebart, James Andrew Bagnell, and Martial Hebert. Activity forecasting. In *European Conference on Computer Vision*, pages 201–214. Springer, 2012. 2

[23] Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. A hierarchical representation for future action prediction. In *European Conference on Computer Vision*, pages 689–704. Springer, 2014. 2

[24] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 336–345, 2017. 2, 6

[25] Namhoon Lee and Kris M Kitani. Predicting wide receiver trajectories in american football. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016. 2

[26] Yong Jae Lee, Joydeep Ghosh, and Kristen Grauman. Discovering important people and objects for egocentric video summarization. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1346–1353. IEEE, 2012. 2

[27] Yin Li, Zhefan Ye, and James M Rehg. Delving into egocentric actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 287–295, 2015. 2

[28] Wei-Chiu Ma, De-An Huang, Namhoon Lee, and Kris M Kitani. Forecasting interactive dynamics of pedestrians with

fictitious play. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4636–4644. IEEE, 2017. 2
[29] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016. 1, 2
[30] Montiel J. M. M. Mur-Artal, Raúl and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 3, 5
[31] Igor Najfeld and Timothy F Havel. Derivatives of the matrix exponential and their computation. *Advances in applied mathematics*, 16(3):321–375, 1995. 4
[32] Hamed Pirsiavash and Deva Ramanan. Detecting activities of daily living in first-person camera views. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2847–2854. IEEE, 2012. 2
[33] Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning*, pages 729–736. ACM, 2006. 2
[34] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015. 4
[35] Nicholas Rhinehart and Kris M Kitani. First-person activity forecasting with online inverse reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3696–3705, 2017. 2
[36] Nicholas Rhinehart, Kris M Kitani, and Paul Vernaza. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *European Conference on Computer Vision*, pages 794–811. Springer, Cham, 2018. 1, 2, 3, 6
[37] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016. 2
[38] Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014. 2
[39] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011. 2
[40] MS Ryoo, Thomas J Fuchs, Lu Xia, Jake K Aggarwal, and Larry Matthies. 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*, pages 295–302. ACM, 2015. 2
[41] Michael S Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1036–1043. IEEE, 2011. 2
[42] Michael S Ryoo and Larry Matthies. First-person activity recognition: What are they doing to me? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2730–2737, 2013. 2
[43] Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012. 2, 5
[44] Hyun Soo Park, Jyh-Jing Hwang, Yedong Niu, and Jianbo Shi. Egocentric future localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4697–4705, 2016. 2
[45] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–106, 2016. 2
[46] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pages 835–851. Springer, 2016. 6
[47] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016. 4
[48] Xinshuo Weng, Ye Yuan, and Kris Kitani. Joint 3d tracking and forecasting with graph neural network and diversity sampling. *arXiv:2003.07847*, 2020. 2
[49] Dan Xie, Sinisa Todorovic, and Song-Chun Zhu. Inferring "dark matter" and "dark energy" from videos. *2013 IEEE International Conference on Computer Vision*, pages 2224–2231, 2013. 2
[50] Takuma Yagi, Karttikeya Mangalam, Ryo Yonetani, and Yoichi Sato. Future person localization in first-person videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
[51] Ye Yuan and Kris Kitani. 3d ego-pose estimation via imitation learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 735–750, 2018. 2
[52] Ye Yuan and Kris Kitani. Diverse trajectory forecasting with determinantal point processes. *arXiv preprint arXiv:1907.04967*, 2019. 2
[53] Ye Yuan and Kris Kitani. Ego-pose estimation and forecasting as real-time pd control. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10082–10092, 2019. 2
[54] Ye Yuan and Kris Kitani. Dlow: Diversifying latent flows for diverse human motion prediction. *arXiv preprint arXiv:2003.08386*, 2020. 2
[55] Mengmi Zhang, Keng Teck Ma, Joo-Hwee Lim, Qi Zhao, and Jiashi Feng. Deep future gaze: Gaze anticipation on egocentric videos using adversarial networks. In *CVPR*, pages 3539–3548, 2017. 2
[56] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2014. 6
[57] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008. 2