Integrated Learning and Decision Making for Autonomous Agents through Energy based Bayesian Models

Abrham Shiferaw Alemaw, Pamela Zontone, Lucio Marcenaro, Pablo Marin, David Martin Gomez, Carlo Regazzoni
University of Genoa, Via Opera Pia 11A, 16145 Genova, Italy
University Charles III of Madrid, Avda de la Universidad 30, 28911, Leganés, Madrid, SPAIN
Email: abrham.shiferaw.alemaw@edu.unige.it, {pamela.zontone, lucio.marcenaro, carlo.regazzoni}@unige.it,
{pamarinp, dmgomez}@ing.uc3m.es

***Abstract*—Generalizability and interpretability are common terminologies that can be found in today's machine learning algorithm design. Generalizability requires a clear understanding of one's own action (self-awareness) and a robust interaction with the environment (situation awareness). Many current studies are devoted in developing an algorithm that is more robust in generalizing unseen situations while explaining self-action. However, such algorithms are complex and are not yet fully developed to be used in production. Intelligent transportation systems like self-driving cars are one of the emerging technologies that need generalizability and explainability in anomalous conditions. We propose to enhance generalizability and interpretability of a self-driving car model by introducing a novel methodology that fuses multi-sensorial data from proprioceptive and exteroceptive sensors of an agent, coupled in a Hierarchical Dynamic Bayesian Network model, in an Active Inference framework. The developed model has three stages: 1) a lower dimensional unsupervised learning stage, considering odometry and action modalities, carried out by first applying Null Force Filtering and then by applying modified GNG clustering algorithms; 2) a self-supervised higher-dimensional video modality learning stage assisted by the learned odometry vocabularies; and 3) an online model-based active learning in continuous and discrete state spaces, and action spaces, in the Active Inference framework. The developed system is tested using the CARLA simulator environment for localizing interacting agents, and exhibits low error compared to state-of-the-art methods.**

I. Introduction

Representing knowledge and inferring it for a particular course of action, like in a sequential state estimation of an agent, ranges from explicitly programming each step in the environment to learning from expert data acquired through different sensors. Autonomous systems are equipped with proprioceptive (steering, throttle, braking) and exteroceptive sensors (camera, LiDAR) sensors to perceive their state and the surrounding environment. Integrating these sensors also with physiological ones [1–3] further improves the driving behavior and the overall safety. This sensory information can be represented through random variables that take different values at each time step, linked through causal effect approaches in a hierarchical representation schema [4]. Modeling sequential data in generalized states has to couple continuous and discrete time series information hierarchically. This hierarchy allows a better representation of knowledge and eases performance measurement (anomaly detection) of each inference at each level, which is invaluable for continual learning. [5, 6]. Hierarchical Dynamic Bayesian representation and inference networks [7, 8] are common and state of the art methodologies for these types of problems. These networks are generative models based on the Bayes' principle that can generate the posterior conditional probability from the prior information updated through the likelihood information. The posterior information tells us the aggregate information about an uncertain state of an agent conditioned on the prior knowledge updated through observations. In sequential state modeling, this posterior information can be seen as a prior knowledge when new information (observation) is available. Considering generalised states it is possible to create more than two hierarchies representing different but related information.

In this work we learn six Hierarchical Dynamic Bayesian Networks (HDBNs) connected to each other for data representation and inference, as it can be seen in Fig. 3: 1) Video State Network; 2) Odometry State Network; 3) Action State Network; 4) Coupled Video and Odometry Vocabulary Network; 5) Action Vocabulary Network; and 6) Configurator (Coupled Action, Video and Odometry) Vocabulary Network. Among these, the Action State Network, the Action Vocabulary Network and the Configurator Network are the innovative contributions on this work, compared to [9]. Here, more emphasis is given to the active learning network, which has casual effect relations to all the other networks, excluding the configurator network, which has a dictionary relation. The active network, which includes both the action state and action vocabulary nodes, is developed based on the Active Inference framework [10]. Unlike the other five networks, the configurator network is a dictionary composed of labels that connects the super states of action, odometry and video networks (vocabulary networks). Hence the action network does not have a direct causal effect relation with the configurator.

The active inference framework is based on the premise that perception and learning can be understood by minimizing a quantity known as variational free energy (VFE), and that action selection, planning, and decision-making can be understood by minimizing the expected free energy (EFE), which quantifies the VFE of various actions based on expected future outcomes [11]. As discussed in [10], VFE is a resemblance of the capacity of a human brain in doing gradient descent at the time of perception, which can be associated to the mismatch between prediction and update messages. The other quantity, i.e., EFE, models the rule of action selection, which is a policy predicting the sequences of feature states and observations for each possible action (policy),

by computing the expected free energy (EFE) and selecting the policy that has the minimum value of EFE corresponding to those predicted future states and observations. Mathematically, the VFE F_θ can be expressed as:

$$F_\theta = E_{q(s|\theta)}\left[\ln \frac{q(s|\theta)}{p(o,s|\theta)}\right] \tag{1}$$

where s represents the state, o the observation, θ a possible distribution (policy distribution), p the posterior belief, and q the approximate inference of the posterior belief. The EFE G_θ can be instead written as:

$$G_\theta = E_{q(o,s|\theta)}\left[\ln q(s|\theta) - \ln p(o,s|\theta)\right] \tag{2}$$

In this work we follow a different approach than the original one proposed for VFE and EFE [10]. Here, we consider the learning and decision-making processes in the active stage using Coupled Markov Jump Particle Filters detailed in Algorithm 3 to model perception and action selection procedures through VFE and EFE. In the original work, action is treated as a discrete state which is only optimal for simple tasks. We instead model both continuous and discrete action states.

II. Related Works

The procedures to model and estimate the state of interacting agents need to consider their movement trajectory, the intended course of action, and their surrounding environment concisely. The viewpoint towards these interaction models should be designed and configured as a natural process of behavioral decision-making methodologies in which a new state is at least partially predictable by utilizing the current state and plausible actions conforming to the evidence (observations) [12]. Designing and developing such solutions should focus on how to guide the learning process through prediction errors (mismatches between expected and observed outcomes) corrected through self-actions as a replication of biological learning and decision-making process.

A. Interaction and Localization

Studies have been proposed to mitigate interaction problems for pedestrians [13] and vehicles [14]. Other interaction models can oversee the interaction between ego agents and pedestrians [15]. Dynamic Bayesian Network (DBN) models are also demonstrated to be effective in anomaly detection [16] using the adapted version of the Markov Jump Particle Filter [17]. The focus of these studies is mainly on optimizing anomaly prediction while predicting the trajectory of a single ego vehicle. Even if they can better detect and interpret why anomalies are present, those models did not consider how these interactions can be modeled at the time of online decision-making.

Probabilistic Graphical Models (PGMs), particularly DBNs, have the advantage of coherently predicting intentions and trajectories for pedestrians and vehicles at the same time [18]. They are inherently interpretable due to the causal structure with variables representing explicit physical meanings through the integration of different sensory inputs. DBNS are generalizable in representing and inferring heterogeneous signals in time-series data modeling.

B. Active Learning and Inference

Deep Learning based autonomous driving models proposed in literature mainly use Reinforcement Learning [19], Imitation learning [20], Active Inference [10], and a hybrid combination of these frameworks. The hybrid models are proposed to account for expert data insufficiency, minimizing the number of explorations [21] and enhancing explainability [22].

Imitation learning (IL) is a mechanism to learn a model offline and to use this model in online decision-making tasks. IL, in contrast to RL, uses expert knowledge to discriminate non-optimal actions without trial-and-error runs to get feedback from the environment. However, IL models suffer from cascading errors and distribution shifts since their models are trained only on a subset of the necessary samples (observation-action pairs). Considering self-driving vehicles, in this learning scenario, when these agents encounter slightly different lanes, they shift slightly to the left or right side of the road. This shift feedbacks new observations having more shifts to their models, disturbing their policy until no valid action can be taken anymore.

Reinforcement Learning (RL) is a process of training through an evolutionary method where an agent learns through trial and error actions to interact with the environment. This interaction provides a reward signal indicating the quality of the solution found. Deep RL methods interact with the environment and adjust driving strategies based on environmental feedback. However, Deep RL models [23] require numerous interactions with the environment to ensure that the optimal policy is explored, which is less data efficient. Ideally, these methods can learn the optimal driving policy, but for an autonomous driving task in continuous action space, it requires an agent to spend a lot of time interacting with the environment to explore the optimal policy, and this is not efficient.

The incorporation of observations to action rules through energy-based models is a characteristic of the Active inference framework [10, 11]. In active inference, perception and action selection are inherently unified solutions. The state estimation and the resulting behavior can be harmoniously coupled in a unified framework as a minimization of VFE and EFE. The first step in active inference is to execute a pragmatic action based on the inferred state and observation that fulfills goals directly (i.e., exploitation), followed by performing an epistemic action (i.e., exploration), which discloses information that enables one to choose nonrandom action in the long run (future).

III. Methodology

The learning algorithm starts from an intensive offline learning procedure considering a given set of data. These include video data acquired from an onboard sensor (First Person Viewpoint - FPV), the corresponding odometry data, and the sequences of action data from a moving vehicle driven by an expert driver. The proposed method consists of three stages (see Fig. 4): stage 1 includes the lower dimensional odometry and action vocabulary learning; stage 2 the higher dimensional vocabulary learning; and stage 3 the active learning stage.

A. Odometry Learning

The general assumption in odometry and action learning is to consider the generalized coordinates, which are a set of parameters that can specify the configuration of the dynamics of

Algorithm 1: Action Selection:

action selection is based on epsilon-greedy policy
$\alpha_t = \max_n W_{t,n}$
$\delta_1 \leftarrow threshold$ (0.25) based on odometry anomaly
$\delta_2 \leftarrow threshold$ (0.4) based on video anomaly
$\epsilon_t = 1 - \alpha_t$
if $\epsilon_t < \delta_1$ **then**
    $n_{selected} \leftarrow arg\min G^{(u_i)}_{\theta_{t,n}} Eq.$ (14)
    $\tilde{a}^{(u_i)a}_t = \tilde{a}^{(u_i)a}_{t,n_{selected}}$
else
    if $\epsilon_t < \delta_2$ **then**
        Sample from replay memory: $R(x^{(u_i)v}, x^{(u_i)o}, a^{(u_i)a})$
        Predict states using Algorithm Algorithm 3 using the sampled states
        $n_{selected} \leftarrow arg\min G^{(u_i)}_{\theta_{t,n}} Eq.$ (14)
        $\tilde{a}^{(u_i)a}_t = \tilde{a}^{(u_i)a}_{t,n_{selected}}$
    else
        Uniform sample based on min and max action space of the agent
        $\tilde{a}^{(u_i)a}_t \leftarrow Random(\max_a, \min_a)$
    end if
end if
Environment and action set up (see algorithm Algorithm 2)
$next_{obs}, next_{odom}, reward, done = env.step(action)$
$update_{ReplayMemory}(state, odom, action, nextState, nextOdom, reward)$

Algorithm 2: Environment Configuration:

$control_{throttle} = action_{throttle}$
$control_{steer} = action_{steer}$
$control_{brake} = action_{brake}$
Acceleration regulation:
if $action_{throttle} > 0$ **then**
    $control_{throttle} = min(control_{throttle}, maxThrot)$
    $control_{brake} = 0.0$
else
    $control_{throttle} = 0.0$
    $control_{brake} = min(abs(control_{throttle}), maxBrake)$
end if
Steering regulation:
if $control_{steer} > 0$: **then**
    $control_{steer} = \min(maxSteer, control_{steer})$
else
    $control_{steer} = \min(-maxSteer, control_{steer})$
end if
$vehicle_{control}(control)$
$vel = vehicle_{velocity}$
$Kmh = 3.6 * \sqrt{vel.x^2 + vel.y^2 + vel.z^2}$
if $collision$: **then**
    done = True
    reward = -1
else
    $done = False$
    $reward = 1$
end if
if $SECONDS\_PER\_EPISODE < time.time()$: **then**
    $done = True$
end if
Output: image, vehicle location, reward, done

a physical system as a time derivative of its state. Generalized coordinates are used to develop the equation of motion of a system as a function of time. Generalized coordinates are simplifications of the motion equation in the assumption of Lagrangian mechanics with the principle of least action [24]. In generalized coordinate systems, having positional stationary points $[x_t, y_t]$, one can apply the Null Force Filter [25], which computes the n_{th} order derivatives of the positional data. In this paper for odometry learning, we only used the first-order derivative. The sensory observations of the odometry module comprise the positional information of the agent along axis x and y such as $z^o_{t,u_i} = [\mathrm{x}_{t,u_i}, \mathrm{y}_{t,u_i}]$, with i representing two interacting agents (unmanned vehicle1 and unmanned vehicle2) overtaking each other. The odometry generalized state comprises position and velocity of each agent, i.e., $\tilde{z}^o_{t,u_i} = [\mathrm{x}_{t,u_i}, \mathrm{y}_{t,u_i}, \mathrm{v}_{\mathrm{x}_{t,u_i}}, \mathrm{v}_{\mathrm{y}_{t,u_i}}]$. We further process the GSs to have a semantic category (cluster) depicting the movement modality of both agents using the Modified Growing Neural Gas (M-GNG) algorithm [26]. The eight-dimensional feature vector $\mathcal{F}^o_{t,u_1}$ w.r.t. u_1 is given as an input to the clustering algorithm, which can be written as:

$$\mathcal{F}^o_{t,u_1} = [\mathrm{x}_{t,u_1}, \mathrm{y}_{t,u_1}, \mathrm{v}_{\mathrm{x}_{t,u_1}}, \mathrm{v}_{\mathrm{y}_{t,u_1}}, \triangle\mathrm{x}_{t,u_{12}}, \triangle\mathrm{y}_{t,u_{12}}, \triangle\mathrm{v}_{\mathrm{x}_{t,u_{12}}}, \triangle\mathrm{v}_{\mathrm{y}_{t,u_{12}}}], \quad (3)$$

where $\{\triangle\mathrm{x}_{t,u_{12}}, \triangle\mathrm{y}_{t,u_{12}}\}$ and $\{\triangle\mathrm{v}_{\mathrm{x}_{t,u_{12}}}, \triangle\mathrm{v}_{\mathrm{y}_{t,u_{12}}}\}$ are the relative positions and velocities of the interacting vehicle u_1 and neighbouring vehicle u_2, respectively.

For each cluster $\tilde{s}_k$, where $k = 1 \ldots K$, we learn the following cluster or superstate vocabularies: *i)* odometry cluster mean $M^{(\tilde{s}_k)O}$; *ii)* odometry cluster covariance $Q^{(\tilde{s}_k)O}$; *iii)* odometry transition matrix T defining the probability of transition from one cluster to the other; *iv)* odometry transition time $T^{(g)}$ defining the time spent in each cluster before switching (moving) to the next cluster. These properties are learned through M-GNG in an unsupervised machine learning way. The learned HDBN (see the odometry vocabulary in Fig. 3) is a hierarchical generative model that allows us to represent the learned probabilistic conditional distribution of each agent between two consecutive time instants, $t-1$ and t (previous state and current state), so that we can predict the next state $t+1$ in a semi-Markovian model.

B. Action Learning

In stage one, the second network, which concerns the action vocabulary learning, is learned from the output of control information. This comprises throttle, steering angle, and brake information. The action datasets are extracted from the CARLA [27] simulator which is programmed to record the action information being sent to the actuators of the interacting agents. One thing to note here is that the video, odometry, and action datasets are recorded from this simulator being synchronized to record this information. This means that, at each point in time, it records a video frame, an odometry point, and the action taken at that point.

The sequence of action data is clustered using the M-GNG algorithm to account for the discrete action space. This allows us to create the action vocabulary which includes *i)* action cluster mean $M^{(\tilde{s}_k)a}$; *ii)* action cluster covariance $Q^{(\tilde{s}_k)a}$; *iii)* action transition matrix T defining the probability of transition from one cluster to the other; *iv)* action transition time $T^{(g)}$ defining the time spent in each cluster before switching (moving) to the next cluster.

Learning lower dimensional data helps the process of learning the higher dimensional video data to establish a relation at the semantic label. The video cluster learning is guided by the odometry vocabulary that generates a unified network for both video and odometry modalities.

C. Video Learning

The video vocabulary learning model is controlled by the odometry vocabulary to focus the attention of learning towards the best features that represent both the content of the video and the motion, at each consecutive steps.

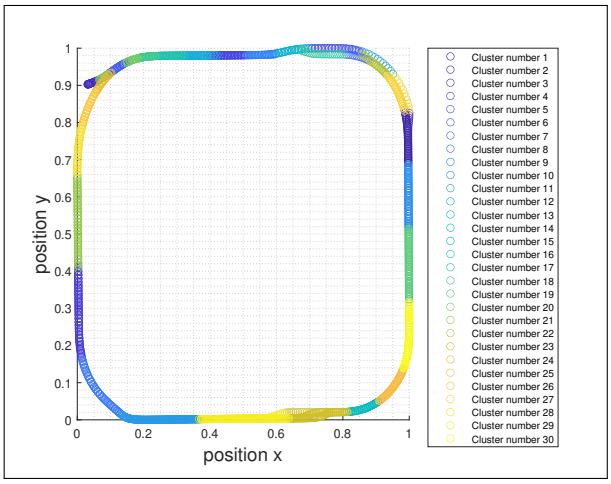

Fig. 1: Clustered trajectory of an overtaking agent showing the output of odometry cluster information.

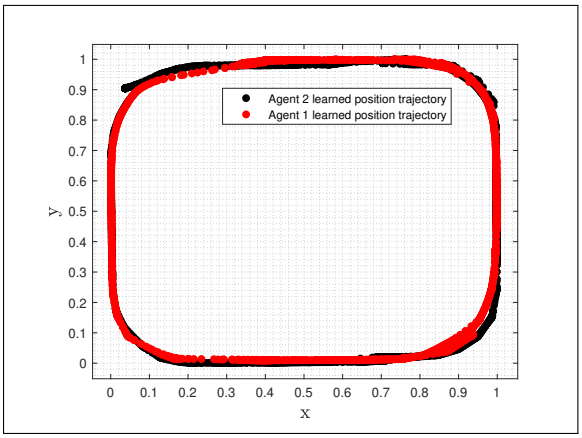

Fig. 2: Learned trajectory from stage 2 (video learning).

After the learning step involving the odometry modality, we use the learned vocabulary to train the video models. For this purpose, the Kalman Variational Autoencoder (KVAE) [28] is employed to obtain the latent states $a_t^{(u_i)v}$ corresponding to the video of the training data $x_t^{(u_i)v}$. To elaborate more, we have two hidden states, i.e., $a_t^{(u_i)v}$ and $z_t^{(u_i)v}$, which represent the content of an image and its corresponding dynamics, respectively. While training, since our main objective is to localize both agents only from the video sequence at each time instant, we learned other pseudo observation matrices, i.e., D and E, from the latent distribution and the odometry vocabulary. These matrices encode the information of the location parameters of interacting agents; therefore, we used four matrices (A, C, D, and E) representing our complex model. The trained odometry vocabulary $\tilde{s}_k^{(u_i)O}$ is employed to guide the model to assign a semantic category to the latent states. The two latent states are modeled as follows:

$$a_{t,u_i} = \sum_{k=1}^{K} \alpha_{t,k,u_i} \times C_{k,u_i} \times z_{t,u_i} + v_{t,u_i}, \tag{4}$$

where a_{t,u_i} is the content distribution that is obtained from the dynamical latent state z_{t,u_i} through a pseudo-observation model. On the other hand, z_{t,u_i} at consequent time instants are connected through the following dynamical model:

$$z_{t+1,u_i} = \sum_{k=1}^{K} \alpha_{t,k,u_i} \times A_{k,u_i} \times z_{t,u_i} + \omega_{t,u_i}, \tag{5}$$

In Eq. (4) and in Eq. (5), v_t and ω_t are Gaussian noises; matrices $\{C_{k,u_i}\}_{k=1\ldots K}$ and $\{A_{k,u_i}\}_{k=1\ldots K}$ represent a set of pseudo-observation models and transition models, respectively, as defined in the traditional Kalman Filter (KF) [29]. These models are combined through a probabilistic vector $\alpha_{t,u_i}^{\tilde{s}_k}$ as a *guiding vector*, which guides the model to assign a semantic category (cluster) to the dynamic video state z_{t,u_i}. Mathematically $\alpha_{t,u_i}^{\tilde{s}_k}$ can be defined as follows:

$$1/\alpha_{t,u_i}^{\tilde{s}_k} = \left(d_{t,u_i}^{\tilde{s}_k}\right)^n \times \sum_{k=1}^{K} \frac{1}{\left(d_{t,u_i}^{\tilde{s}_k}\right)^n}, \tag{6}$$

where $d_{t,u_i}^{\tilde{s}_k} = \mathcal{D}_M((z_{t,u_i}^v), (M^{(\tilde{s}_k)o}, Q^{(\tilde{s}_k)o}))$ is the Mahalanobis distance ($\mathcal{D}_M$) [30] calculated between each latent state distribution z_{t,u_i}^v and the probability distribution of each cluster of the odometry module $(M^{(\tilde{s}_k)o}, Q^{(\tilde{s}_k)o})$. $M^{(\tilde{s}_k)o}$ is the mean of cluster $\tilde{s}_k$, and $Q^{(\tilde{s}_k)o}$ is its covariance, i.e., the probability distribution is Gaussian. In the first iteration, z_{t,u_i}^v will be assigned to the superstates corresponding to the shortest distance $d_{t,u_i}^{\tilde{s}_k}$ values. After the first iteration, $d_{t,u_i}^{\tilde{s}_k}$ will be multiplied by the previous $\alpha_{t,u_i}^{\tilde{s}_k}$ value to cluster the z_{t,u_i}^v, having similar probability distributions.

Similarly the video training can be guided, using the action vocabulary, through a probabilistic guiding vector:

$$1/\beta_{t,u_i}^{\tilde{j}_k} = \left(d_{t,u_i}^{\tilde{j}_k}\right)^n \times \sum_{k=1}^{K} \frac{1}{\left(d_{t,u_i}^{\tilde{j}_k}\right)^n}, \tag{7}$$

where $d_{t,u_i}^{\tilde{j}_k} = \mathcal{D}_M((z_{t,u_i}^v), (M^{(\tilde{j}_k)a}, Q^{(\tilde{j}_k)a}))$ is the Mahalanobis distance ($\mathcal{D}_M$) between the video latent state and the mean of the action cluster.

In [28], after performing Kalman smoothing, four main losses are used to optimize the model:

(a) the traditional reconstruction loss $\mathcal{L}_{recon}$ (between $x_t^{(u_i)v}$ and its VAE reconstruction $\hat{x}_t^{(u_i)v}$);

(b) the Kullback Leibler Divergence term of the VAE, together with $\mathcal{L}_{recon}$, represents the total VAE loss $\mathcal{L}_{VAE}$;

(c) a transition loss $\mathcal{L}_{trans}$ enforcing the learning of the transition models A using Bhattacharya distance (DB):

$$\mathcal{L}_{trans} = D_B(zs_{t+1}, Azs_t) \tag{8}$$

(d) an emission loss $\mathcal{L}_{emiss}$ enforcing the learning of the emission models C using D_B:

$$\mathcal{L}_{emiss} = D_B(a_t, Czs_t) \tag{9}$$

While training, we also learned the *pseudo observation vectors* $\hat{z}_{t,u_1}^o, \hat{z}_{t,u_2}^o$ as follows:

$$\hat{z}_{t,u_1}^o = D^{\tilde{s}_k} \times z_{t,u_1}^o + E^{\tilde{s}_k} + M^{\tilde{s}_k^O} + e_{1_t}, \tag{10}$$

$$\hat{z}_{t,u_2}^o = z_{t,u_1}^o + r_{d_t,u_{12}} + e_{2_t}, \tag{11}$$

where the prediction of $\hat{z}_{t,u_1}^o$ and $\hat{z}_{t,u_2}^o$ are performed by minimizing the errors (e_{1_t}, e_{2_t}) w.r.t. the ground truth odometry

(see Fig. 1). Optimization is performed through minimization of the following losses:

$$L_{\delta,u_1} = \begin{cases} \frac{1}{2}(\tilde{z}_{t,u_1} - \hat{z}_{t,u_1})^2 & if \left|(\tilde{z}_{t,u_1} - \hat{z}_{t,u_1})\right| < \delta \\ \delta(|\tilde{z}_{t,u_1} - \hat{z}_{t,u_1}| - \frac{1}{2}\delta) & otherwise \end{cases}, \tag{12}$$

$$L_{\delta,u_2} = \begin{cases} \frac{1}{2}(\tilde{z}_{t,u_2} - \hat{z}_{t,u_2})^2 & if \left|(\tilde{z}_{t,u_2} - \hat{z}_{t,u_2})\right| < \delta \\ \delta(|\tilde{z}_{t,u_2} - \hat{z}_{t,u_2}| - \frac{1}{2}\delta) & otherwise \end{cases}, \tag{13}$$

where L_{δ,u_1} and L_{δ,u_2} are the *Huber Loss functions* [31] with $\delta = 1$. During the (online) testing phase, the learned vocabularies s^v_{k,u_1}, $\tilde{s}^o_{k,u_1}$ and $\hat{z}^o_{t,u_1}$, $\hat{z}^o_{t,u_2}$ are employed to predict the trajectories of the interacting agents.

D. Learned HC-DBN Model

As it can be seen in Fig. 3, the inference related to the perception model can be carried out in two ways. The first one is the prediction from the same state in different time steps, meaning from $time_t$ to $time_{t+1}$ for each state and super state (from perceptual state to perceptual state at a different time), while the second inference is the prediction from the action state at the same time instant (the effect of the predicted action on the perceptual states). These two ways of inferences can give the learned model the chance to have different signals for anomaly measurements from multi-level inference mechanisms. This can be useful to select an action by checking the impact of the selected action on the current states.

In our models, to integrate action and perception models, we use the Multi-Agent Combined Markov Jump Particle Filters (MAC-MJPFs). We chose these filters because they are integrated and well-suited to adapt particle estimations by fast reinitialization when the agent changes its motion. As a single integrated multi-filter, this filter allows updating each estimate after each measurement to get an updated state and state covariance, by combining the most likely models based on their weights.

E. Active Learning and Inference

In this stage, which is the last one, the agent is equipped with an environmental model from the learned vocabularies. Inferring and integrating these learned models with decision-making using Active inference are modeled through a Coupled Markov Jump Particle Filters (Algorithm 3).

Given the input of both vocabularies, the agent sends video information to the active learning model. The model first gets the video latent state and initializes each super state (S_t, J_t and D_t) based on the probabilistic distance vector (α_t). After the initialization, it predicts the next super states using the particle filter, and the next continuous states using the Kalman filter. It then updates each prediction after the action selection. After the update phase, since we are using a particle filter to account for particle degeneracy problem [32], we apply a resampling step with a replacement, based on a threshold, after each update step. More specifically, we do not resample after each update step if the number of particles that have nearly zero weight is less than 25% of the total number of particles. This allows us to improve the overall performance.

Algorithm 3: AIF for Interaction:

Input: $M^{(\tilde{s}_k)^o}, Q^{(\tilde{s}_k)^o}, T, T^{(g)}, M^{(s_k,a_t)^v}, M^{(s_k,z_t)^v}$
$Q^{(s_k,a_t)^v}, Q^{(s_k,z_t)^v}, x_t^{(u_1)^v}, M^{(\tilde{j}_k)^a}, Q^{(\tilde{j}_k)^a}$
for $t = 1, \ldots, \tau \leftarrow$ *Time evolution* **do**
  Obtain image latent state using VAE: $a_t^{(u_1)^v}$, $\sigma_t^{(u_1)^v}$ and
  Obtain latent state covariance
  Calculate probabilistic *guiding vector*
  Begin Filtering
  if $t == 1$ **then**
    INITIALIZATION OF PARTICLES:
    for $n = 1, \ldots, N \leftarrow$ *number of Particles* **do**
      Initialize cluster assignment $\tilde{s}_{t=1,n}$ of particle n
      using cluster probabilities $\alpha_{v,t}^{(\tilde{s}_k^{u_i})}$ with $\tilde{s} = 1, \ldots, K$
      Initialize n^{th} particle video value as:
      $z^v_{t=1,n} \sim \mathcal{N}(M^{(s_{1,n},z)^v}, Q^{(s_{1,n},z)^v})$
      Initialize n^{th} particle odometry value as:
      $z^o_{t=1,n} \sim \mathcal{N}(M^{(s_{1,n},z)^o}, Q^{(s_{1,n},z)^o})$
      Initialize n^{th} particle action value as:
      $a_{t=1,n} \sim \mathcal{N}(M^{j_{1,n},a}, Q^{j_{1,n},a})$
      Initialize n^{th} particle configurator value as:
      $c_{t=1,n} \sim \mathcal{N}(M^{c_{1,n},c}, Q^{c_{1,n},c})$
    end for
  else
    UPDATE:
    for $n = 1, \ldots, N \leftarrow$ *number of Particles* **do**
      Perform video update:
      $z^{(u_i)^v}_{t|t,n}, \Sigma^{(u_i)^v}_{t|t,n} =$
      $KF^{Video}_{update}(z_{t|t-1,n}, \Sigma_{t|t-1,n}, C^{s_t}_{t,n}, a_{t,n}, \mathcal{R}^v_{t,n})$
      Obtain odometry estimation from video:
      $\tilde{z}^{(u_i)^o}_{t|t,n} = D^{(\tilde{S}^O,O)} * z_t + E^{(\tilde{S}^O,O)} + M^{(\tilde{S}^O,O)}$
      Perform odometry update:
      $\tilde{z}^{(u_i)^o}_{t|t,n}, \Sigma^{(u_i)^o}_{t|t,n} =$
      $KF^{Odom}_{update}(\tilde{z}_{t|t-1,n}, \Sigma_{t|t-1,n}, \hat{z}_{t|t-1,n}, \mathcal{R}^o_{t,n})$
      Perform action update:
      $\tilde{a}^{u_i}_{t|t,n}, \Sigma^{u_i}_{t|t,n} =$
      $KF^{Action}_{update}(\tilde{a}_{t|t-1,n}, \Sigma_{t|t-1,n}, \hat{a}_{t|t-1,n}, \mathcal{R}^a_{t,n})$
    end for
    Calculate anomalies.
    Re-weight v,o,a particles weights based on anomalies
    Re-sample if:
    $\sum_n 1/w_n^2 <$ threshold (we used 0.25)
  end if
  PREDICTION:
  for $n = 1, \ldots, N \leftarrow$ *number of Particles* **do**
    Predict vocabulary of video and odometry:
    $s^{(u_i)^{v,o}}_{t+1,n} = s^{(u_i)^{v,o}}_{t,n} | s^{(u_i)^{v,o}}_{t,n} \sim T(s^{v,o}_{t,n})$
    Predict vocabulary of action:
    $J^{(u_i)^a}_{t+1,n} = J^{(u_i)^a}_{t+1,n} | J^{(u_i)^a}_{t,n} \sim T(J^a_{t,n})$
    Predict vocabulary of configurator:
    $D^{(u_i)^c}_{t+1,n} = D^{(u_i)^c}_{t+1,n} | D^{(u_i)^c}_{t,n} \sim T(D^c_{t,n})$
    Predict Video state:
    $z^{(u_i)^v}_{t+1|t,n}, \Sigma^{(u_i)^v}_{t+1|t,n} = KF^V_{pred}(z^{(u_i)^v}_{t|t,n}, \Sigma^{(u_i)^v}_{t|t,n}, s^{(u_i)^{v,o}}_{t+1,n})$
    Predict odometry state:
    $\tilde{z}^{(u_i)^o}_{t+1|t,n}, \Sigma^{(u_i)^o}_{t+1|t,n} = KF^O_{pred}(z^{(u_i)^v}_{t|t,n}, \Sigma^{(u_i)^v}_{t|t,n}, \tilde{s}^{(u_i)^{v,o}}_{t+1,n})$
    Predict action state:
    $\tilde{a}^{(u_i)^a}_{t+1|t,n}, \Sigma^{(u_i)^a}_{t+1|t,n} = KF^A_{pred}(a^{(u_i)^a}_{t|t,n}, \Sigma^{(u_i)^a}_{t|t,n}, D^{(u_i)^c}_{t+1,n})$
    $\tilde{a}^{(u_i)^a}_{t+2|t+1,n}, \Sigma^{(u_i)^a}_{t+2|t+1,n} =$
    $KF^A_{pred}(a^{(u_i)^a}_{t+1|t+1,n}, \Sigma^{(u_i)^a}_{t+1|t+1,n}, D^{(u_i)^c}_{t+2,n})$
  end for
  Action selection: See Algorithm 1
end for
Output: Optimal Policy

Action selection (Algorithm 1) is using the free energy principle. To select an action, the configurator network should first combine the predicted action super state, and the predicted combined odometry and video super states, to select the most probable state. Then it can select the higher weighted particle and associate it with the continuous action. This maximum weight is used to choose between exploration (new actions) and exploitation (the learned action).

In active inference, to select an action and update the states,

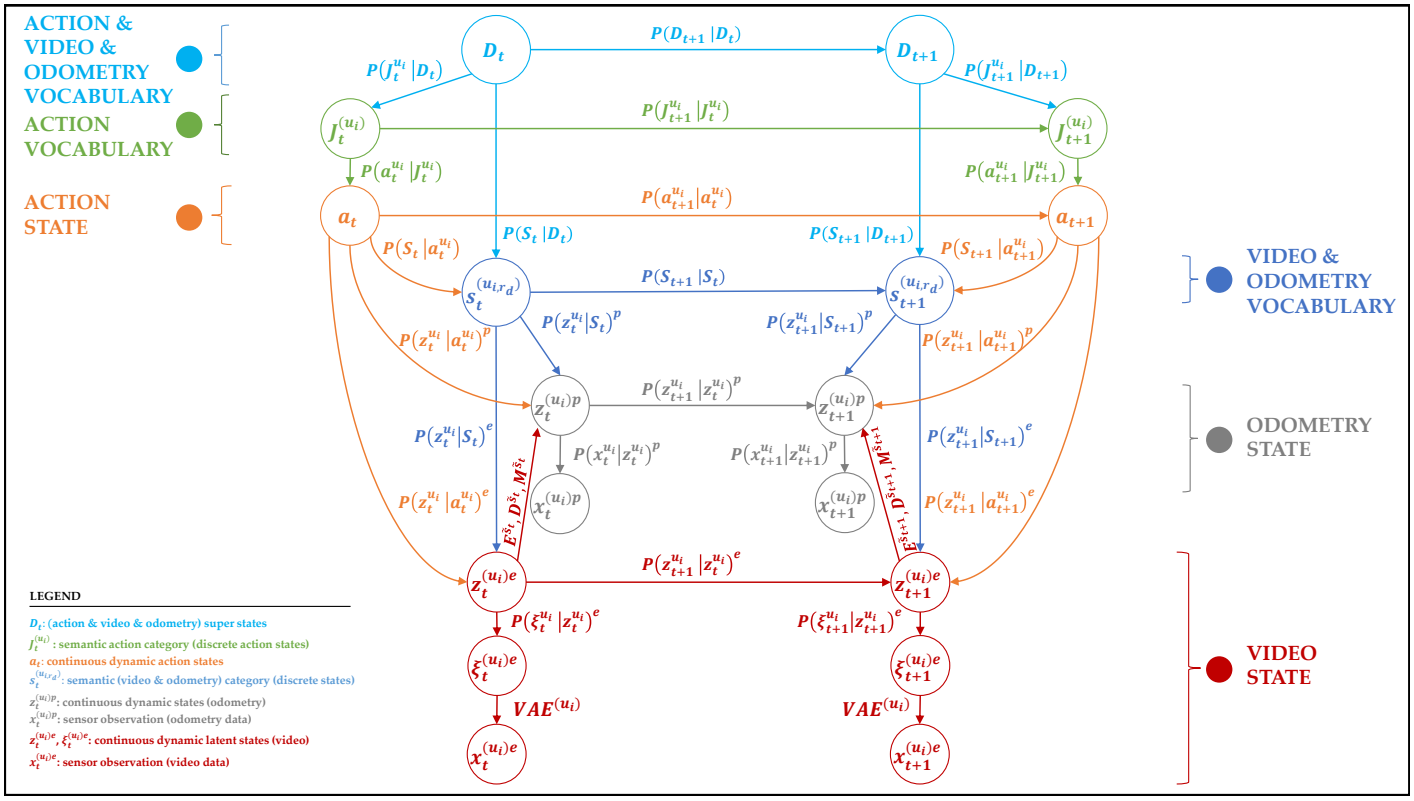

Fig. 3: Learned HC-DBN.

Friston *et al*. [10] used, as an EFE, the sum of the scalar product between the logarithmic value of the preferred observation and the predicted state. In our case, we use the learned trajectory from Sec. III-C as a preferred state and observation, and for each prediction of the particles, we measure the Mahalanobis distance between the preferred generalized state and the predicted generalized state:

$$G_{\theta_{t,n}}^{(u_i)} = D_M(\hat{z}_t^{(u_i)o}, \tilde{z}_{t,n}^{(u_i)o}) \tag{14}$$

where N is the total number of particles at that time instant. To select the action, we use the minimum index of $G_{\theta_{t,n}}^{(u_i)}$ and select that particle from the predicted action particles.

IV. Results

Stage one and stage two of the model are trained offline. From these trained models the agent should be able to navigate the learned environment (Fig. 2) with minimal error. Tab. I shows a comparison with state-of-the-art methods. Tab. II shows generalized state estimation RMSE drifts between the interacting agents. Sequences of action while exploring the environment are shown in Fig. 6.

TABLE I: Table displaying the comparison of Root Mean Square Error (RMSE) drifts of different methods.

Methods	RMSE
UnDeepVO [33]	4.07
DSV-SLAM [34]	3.693
Frame to Frame correspondence [35]	5
ILDM (ours)	**2.897**

TABLE II: It displays the quantitative analysis of the overtaking model tested with two (01 and 02) overtaking experiments, using RMSE measurements. We used the generalized training data as a ground truth to compute the errors.

RMSE	x	y	v_x	v_y	Exp No.
Overtaking a1	1.7814	1.7560	0.8693	0.9646	01
Overtaking a1	1.5367	4.1848	0.8758	0.8644	02
Overtaking a2	1.7550	3.1502	0.0052	0.0049	01
Overtaking a2	1.7218	1.0127	0.0058	0.0184	02

Considering the selection of the sequence of actions, the agent is able to take actions (decisions) that minimize the EFE. Minimizing the EFE gives the agent the chance to minimize the VFE at the perception level. Hence, good policy inference is a starting point to perform well in perception during the task of interaction. Further, the model predicted well the anomalous incidents, by taking explainable decisions (Fig. 5).

DSV-SLAM is a LiDAR descriptor-based SLAM approach proposed in [34] to facilitate an efficient detection of loop closures for localization and mapping. They used the KITTI dataset and achieved an RMSE of 3.693% on average. Other methods, like UnDeepVO [33], perform pose estimation by training deep neural networks in an unsupervised manner, and attain an average of 4.07% RMSE drift. Li *et al*. [35] proposed a self-supervised learning framework to represent frame-to-frame correspondence for depth and pose estimation with RMSE drift of around 5%. These comparisons show that our method allows us to estimate the trajectories of interacting agents with lower RMSE drifts of 2.897%.

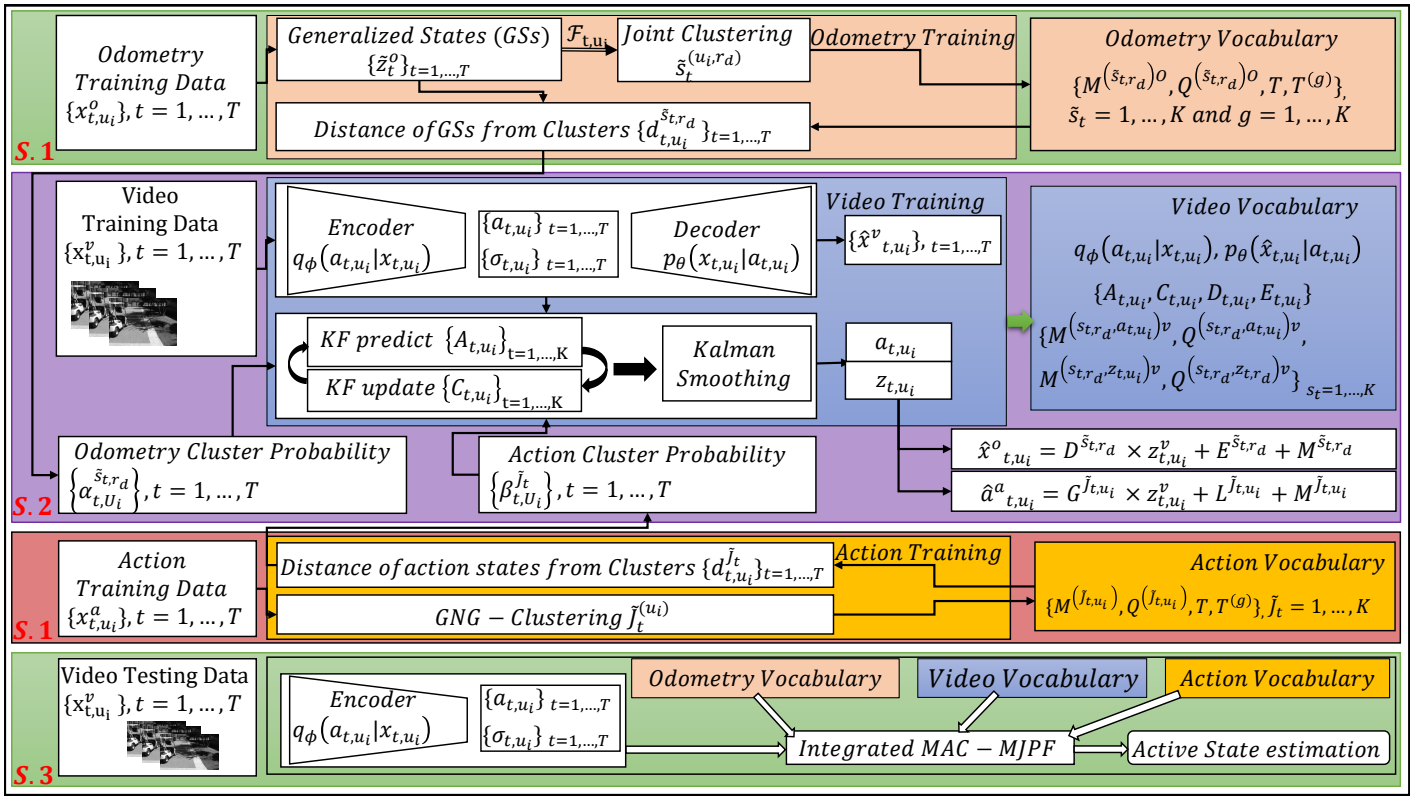

Fig. 4: Integrated self-representation learning and decision-making from exteroceptive information driven by learned vocabulary of proprioceptive information.

Fig. 5: Explaining different sequences of actions in terms of anomaly signals.

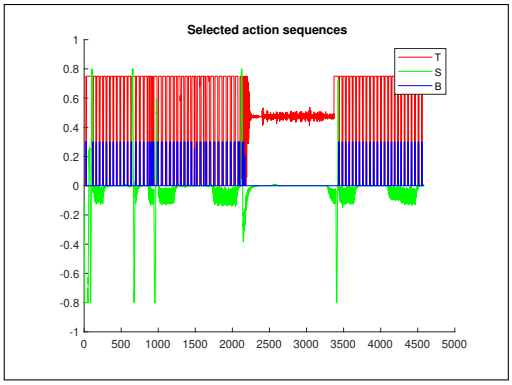

Fig. 6: Throttle, steering angle, and brake sequences of online predicted actions.

V. Conclusion and Future Work

This paper proposed a data-driven approach for the Coupled Bayesian learning and decision-making of interacting agents through multi-sensorial data. During offline training, the vocabulary of the low-dimensional modality is used to learn the latent states and their dynamics for the video modality, obtaining in this way the corresponding vocabulary. During online testing, the trajectories of the interacting agents are inferred only from the video sensory signals and the learned vocabularies. Our experiments showed that this knowledge allows an agent to localize itself and navigate the environment by minimizing the expected free energy, in a self-supervised approach. For future work, we plan to extend the proposed integrated Bayesian learning and decision-making model for the localization and interactions of heterogeneous agents.

VI. Acknowledgement

This research was partially funded by the European Union's Horizon Europe research and innovation programme under the Grant Agreement No. 10112113, and by the European Union - NextGenerationEU and by the Ministry of University and Research (MUR), National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.5, project "RAISE - Robotics and AI for Socio-economic Empowerment" (ECS00000035).

References

[1] P. Zontone *et al.*, "Stress recognition in a simulated city environment using skin potential response (SPR) signals," in *2021 IEEE International Workshop on Metrology for Automotive (MetroAutomotive)*, 2021, pp. 135–140.

[2] T. Aminosharieh Najafi *et al.*, "Driver attention assessment using physiological measures from EEG, ECG, and EDA signals," *Sensors*, vol. 23, no. 4, 2023.

[3] ——, "Drivers' mental engagement analysis using multi-sensor fusion approaches based on deep convolutional neural networks," *Sensors*, vol. 23, no. 17, 2023.

[4] C. S. Regazzoni *et al.*, "Multisensorial generative and descriptive self-awareness models for autonomous systems," *Proceedings of the IEEE*, vol. 108, no. 7, 2020.

[5] G. Slavic *et al.*, "A kalman variational autoencoder model assisted by odometric clustering for video frame prediction and anomaly detection," *IEEE Transactions on Image Processing*, vol. 32, pp. 415–429, 2023.

[6] M. Ravanbakhsh *et al.*, "Learning self-awareness for autonomous vehicles: Exploring multisensory incremental models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3372–3386, 2021.

[7] C. Cappelle *et al.*, "Multi-sensors data fusion using dynamic bayesian network for robotised vehicle geo-localisation," *2008 11th International Conference on Information Fusion*, pp. 1–8, 2008.

[8] P. Han *et al.*, "Active and dynamic multi-sensor information fusion method based on dynamic bayesian networks," *2009 International Conference on Mechatronics and Automation*, pp. 3076–3080, 2009.

[9] A. S. Alemaw *et al.*, "A data-driven approach for the localization of interacting agents via a multi-modal dynamic bayesian network framework," in *2022 18th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2022, pp. 1–8.

[10] K. J. Friston *et al.*, "Active inference and epistemic value," *Cognitive Neuroscience*, vol. 6, pp. 187 – 214, 2015.

[11] R. Smith *et al.*, "A step-by-step tutorial on active inference and its application to empirical data," *Journal of Mathematical Psychology*, vol. 107, p. 102632, 2022.

[12] D. Marković *et al.*, "Predicting change: Approximate inference under explicit representation of temporal structure in changing environments," *PLoS Computational Biology*, vol. 15, 2019.

[13] D. A. Ridel *et al.*, "A literature review on the prediction of pedestrian behavior in urban scenarios," *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3105–3112, 2018.

[14] S. Lefèvre *et al.*, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH Journal*, vol. 1, no. 1, p. 1, 2014.

[15] G. Slavic *et al.*, "Anomaly detection in video data based on probabilistic latent space models," in *IEEE Conference on Evolving and Adaptive Intelligent Systems*, 2020.

[16] ——, "Multilevel anomaly detection through variational autoencoders and bayesian models for self-aware embodied agents," *IEEE Transactions on Multimedia*, 2022.

[17] M. Ravanbakhsh *et al.*, "Learning self-awareness for autonomous vehicles: Exploring multisensory incremental models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3372–3386, 2021.

[18] G. Slavic *et al.*, "Learning of linear video prediction models in a multi-modal framework for anomaly detection," in *IEEE International Conference on Image Processing*, 2021, pp. 1569–1573.

[19] J. Hu *et al.*, "Autonomous motion decision-making based on deep reinforcement learning for autonomous driving," in *2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, 2022, pp. 1–6.

[20] J. Kim *et al.*, "Advisable learning for self-driving vehicles by internalizing observation-to-action rules," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9658–9667.

[21] Z. Li, "A hierarchical autonomous driving framework combining reinforcement learning and imitation learning," in *2021 International Conference on Computer Engineering and Application (ICCEA)*, 2021, pp. 395–400.

[22] S. Nozari *et al.*, "Active inference integrated with imitation learning for autonomous driving," *IEEE Access*, 2022.

[23] A. Gupta *et al.*, "Safe driving of autonomous vehicles through state representation learning," in *2021 International Wireless Communications and Mobile Computing (IWCMC)*, 2021, pp. 260–265.

[24] K. Friston *et al.*, "Dem: A variational treatment of dynamic systems," *NeuroImage*, vol. 41, no. 3, pp. 849–885, 2008.

[25] H. Iqbal *et al.*, "Data-driven transition matrix estimation in probabilistic learning models for autonomous driving," *Signal Processing*, vol. 188, p. 108170, 2021.

[26] ——, "Clustering optimization for abnormality detection in semi-autonomous systems," 10 2019.

[27] A. Dosovitskiy, G. Ros, F. Codevilla, A. M. López, and V. Koltun, "Carla: An open urban driving simulator," *ArXiv*, 2017.

[28] M. Fraccaro *et al.*, "A disentangled recognition and nonlinear dynamics model for unsupervised learning," in *Conference on Neural Information Processing Systems*, 2017.

[29] M. Ravanbakhsh *et al.*, "Learning self-awareness for autonomous vehicles: Exploring multisensory incremental models," *IEEE Transactions on Intelligent Transportation Systems*, p. 1–15, 2020.

[30] S. Xiang, F. Nie, and C. Zhang, "Learning a mahalanobis distance metric for data clustering and classification," *Pattern recognition*, vol. 41, no. 12, pp. 3600–3612, 2008.

[31] T. Hastie *et al.*, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[32] J. Elfring *et al.*, "Particle filters: A hands-on tutorial," *Sensors (Basel, Switzerland)*, vol. 21, 2021.

[33] R. Li *et al.*, "Undeepvo: Monocular visual odometry through unsupervised deep learning," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7286–7291, 2018.

[34] J. Mo *et al.*, "Fast direct stereo visual slam," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 778–785, 2022.

[35] S. Li *et al.*, "Sequential adversarial learning for self-supervised deep visual odometry," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.