Autonomous Robots
https://doi.org/10.1007/s10514-020-09921-0

Spatially-dependent Bayesian semantic perception under model and localization uncertainty

Yuri Feldman[1] · **Vadim Indelman**[2]

Received: 3 April 2019 / Accepted: 9 June 2020

Abstract
Semantic perception can provide autonomous robots operating under uncertainty with more efficient representation of their environment and better ability for correct loop closures than only geometric features. However, accurate inference of semantics requires measurement models that correctly capture properties of semantic detections such as viewpoint dependence, spatial correlations, and intra- and inter-class variations. Such models should also gracefully handle open-set conditions which may be encountered, keeping track of the resultant model uncertainty. We propose a method for robust visual classification of an object of interest observed from multiple views in the presence of significant localization uncertainty and classifier noise, and possible dataset shift. We use a viewpoint dependent measurement model to capture viewpoint dependence and spatial correlations in classifier scores, showing how to use it in the presence of localization uncertainty. Assuming a Bayesian classifier providing a measure of uncertainty, we show how its outputs can be fused in the context of the above model, allowing robust classification under model uncertainty when novel scenes are encountered. We present statistical evaluation of our method both in synthetic simulation, and in a 3D environment where rendered images are fed into a Deep Neural Network classifier. We compare to baseline methods in scenarios of varying difficulty showing improved robustness of our method to localization uncertainty and dataset shift. Finally, we validate our contribution w.r.t. localization uncertainty on a dataset of real-world images.

Keywords Semantic perception · Object classification and pose estimation · SLAM

1 Introduction

Semantic perception is receiving ever-growing attention in the field of autonomous robotics. In its broad sense semantic perception refers to recovery and understanding of the robot's environment beyond geometric structure. A major line of work has focused on "object-centered" perception Cadena et al. (2016), Salas-Moreno et al. (2013b), which involves recovering poses and classes of objects (or more generally—semantic features) around the robot. While necessary for a robot's interaction with its environment and human operators Sünderhauf et al. (2017), benefits of an object-centered representation of the environment also include memory and computational efficiency, communication efficiency when the said representation needs to be transmitted Choudhary et al. (2017), Salas-Moreno et al. (2013a), when compared to geometric primitives commonly used in SLAM.

Perhaps most importantly, semantic features may be more robustly detected across viewpoints and changes such as in illumination and scale than geometric counterparts, allowing for longer tracking Lianos et al. (2018) and more robust loop closures (e.g. Bowman et al. (2017)), both because of better ability to detect them, and because classification provides good data association priors. Ultimately, focusing on semantic features can provide a manageable number of robust, salient and distinct data association targets.

Distinctness thanks to classification however comes at the condition of its correctness. Misclassifications lead to data association errors, resulting in either duplicate detections

This work was partially supported by the Israel Ministry of Science & Technology (MOST).

✉ Yuri Feldman
yurif@cs.technion.ac.il

Vadim Indelman
vadim.indelman@technion.ac.il

[1] Department of Computer Science, Technion - Israel Institute of Technology, 32000 Haifa, Israel

[2] Department of Aerospace Engineering, Technion - Israel Institute of Technology, 32000 Haifa, Israel

Published online: 25 June 2020

or gross estimate corruption associated with mismatches, making correct classification crucial when coupling pose inference with semantics, thus motivating the fusion of semantic measurements. However, the majority of work so far has focused on object detection and segmentation independently for each new frame, in particular not sharing information across time steps, as opposed to Bayesian fusion of semantic measurements. Rigorous treatment of the latter requires designing classifier noise models, which proves to be nontrivial, as classifier output may vary significantly across viewpoints, particular object instances, imaging conditions such as illumination, partial occlusions and object classes. An important, yet rarely addressed difference from geometrical point features is that semantic measurements are strongly spatially correlated, with nearby views effectively not contributing information w.r.t. one another.

For Machine Learning based classifiers, output scores (features) are further affected by what is generally known as dataset shift (more precisely, covariate shift) Ben-David et al. (2010), Quionero-Candela et al. (2009), i.e. when raw measurements obtained at deployment significantly differ from the ones that the classifier had been trained on. Particularly for Deep Neural Network classifiers, the output in such cases can result in strong response (traditionally interpreted as high confidence), for the wrong class. In our context covariate shift is manifested when encountered scenes are novel w.r.t. the classifier training set, such as objects of novel classes, having unusual appearance, or viewed from a viewpoint not in the classifier training set. While ideally for such a scene non-informative classification is desired, this cannot be guaranteed and in practice is often not the case, since the classifier input is far from the training set. In Bayesian Deep Learning, the distance from the training set is captured by predictive uncertainty, however this information has not been used by previous methods for fusion of classifier measurements.

To cope with the above challenges, many methods propose the use of per-class noise, or rather response, models—rather than directly relying on classifier output. However, commonly, such models are view-independent, capturing both intra-class and viewpoint-dependent variation in classifier outputs as class-dependent noise. In a further simplification, typically most information in classifier output is discarded, save for the most likely class label, further reducing dependence on particular classifier responses and allowing to use a categorical model with Dirichlet priors.

While providing robustness to variations in classifier responses, such models unnecessarily discard information both by not using the entire classifier output vector, and because of overestimating noise by not considering viewpoint-dependence (Fig. 1a). Crucially, not depending on viewpoint renders them unable to capture correlation among similar viewpoints, leading to over-confident inference which does not reflect the actual ambiguity in the measurements.

One challenge to using viewpoint-dependent class models is that the relative pose they rely on is generally unknown, and must be simultaneously estimated. In particular, previous works modeling viewpoint-dependence have either assumed localization is known Atanasov et al. (2014), Teacy et al. (2015), Velez et al. (2012), or resorted to pose space discretization Becerra et al. (2016).

In this paper we develop a method for fusion of classifier measurements focusing on classification of a single object. We propose the following **contributions** w.r.t. previous methods: First, we show how viewpoint - dependent class models capturing both viewpoint—dependent variation and spatial correlation in classifier scores can be applied under uncertain localization with continuous pose. Second, we incorporate semantic measurements carrying information of model uncertainty in inference, allowing classification to be robust to dataset shift, by reflecting the associated uncertainty in the posterior. Finally, our method is able to use the entire classification vector/semantic measurement output by the classifier, making it more general.

The method we describe in this paper was first presented in Feldman and Indelman (2018a, b) and tested in a MATLAB simulation, with classifier measurements generated synthetically. Here we extend the simulation results to more realistic scenarios: first presenting a thorough evaluation in a simulated 3D environment, showing for the first time how spatial viewpoint dependency in the semantic features output by a Deep Neural Network classifier, fed with rendered images, can be modeled and used for object classification under localization and model uncertainty. We then validate our results w.r.t. localization uncertainty on a dataset of real images. Additionally, we analyze and discuss computational aspects.

Akin to the work of Kopitkov and Indelman (2018), who modeled the viewpoint dependency in Deep Learning classifier output to improve pose estimate, our approach uses this viewpoint dependency in a sense in the reverse direction. We present extensive evaluation and parameter sensitivity analysis over scenarios to various degrees affected by localization uncertainty and dataset shift effects.

The rest of the paper is organized as follows. In Sect. 2 we review related work, define the addressed problem in Sect. 3 and the proposed approach in Sect. 4. Section 5 presents the experimental evaluation results. Discussion and conclusions are provided in Sect. 6.

2 Related work

Perception methods performing fusion of classifier measurements can be roughly split into methods directly fusing classifier scores Omidshafiei et al. (2016), Patten et al. (2016), Pillai and Leonard (2015), and methods matching classifier measurements to a statistical model Atanasov et al.

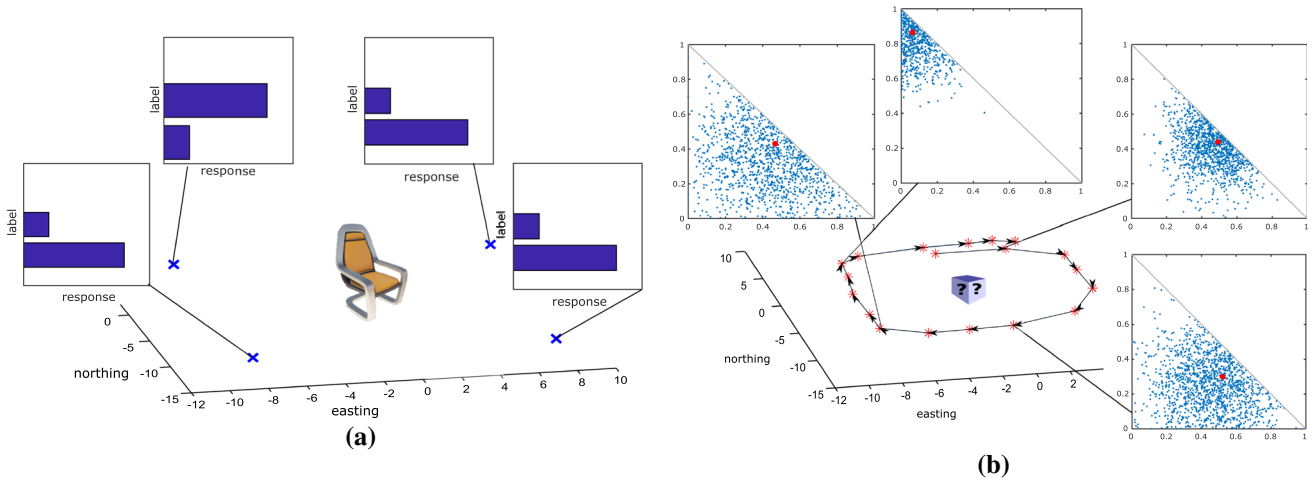


Fig. 1 Left: Viewpoint-dependent variations in classifier outputs are modeled as noise if viewpoint-dependence is ignored. The GP model for an object class (Sect. 3.2) is created from classifier output samples (entire classification vectors) at known relative locations around object. In our simulation scenarios, samples and their locations were specified manually for each of the classes. Right: The robot acquires observations along a track in the vicinity of the object of interest. At each time step, the classifier outputs a cloud of classification vectors reflecting the model uncertainty, unlike a single vector measurement (red dot) or a component thereof in other approaches (Color figure online)

(2014), Becerra et al. (2016), Bowman et al. (2017), Mu et al. (2016), Omidshafiei et al. (2016), Patten et al. (2018), Tchuiev and Indelman (2018), Teacy et al. (2015), Velez et al. (2012). Patten et al. (2016) and Pillai and Leonard (2015) use a classifier unit producing a distribution over classes, i.e. a categorical vector describing class likelihood given an input measurement, fusing measurements by directly applying Bayes rule. Bowman et al. (2017) use most-likely-class measurements with the classifier confusion matrix as the measurement model. In a joint hierarchical Bayesian inference formulation, Omidshafiei et al. (2016) use a Dirichlet class model with categorical vector measurements. In a sense this method belongs to both groups, as only a single noise parameter is inferred per-class, amounting to an assumption that noise-free classifier output is a 1-hot-vector denoting class. Mu et al. (2016) use a Dirichlet class model with a general vector noise parameter with a categorical measurement model (using only the identity of the most - likely class output by the classifier), likewise performing a joint hierarchical inference of states and model parameters. Tchuiev and Indelman (2018) also use a Dirichlet class model with a general noise parameter, incrementally maintaining a Dirichlet classification posterior to quantify model uncertainty.

All methods described above use classifier measurement models that are not parametrized by viewpoint. While this makes class inference trivially agnostic of pose estimation errors, downsides include overly-pessimistic noise model, as variations in measurements due to viewpoint changes are interpreted as noise, and more importantly, over-confident inference due to not accounting for the reduced information value of spatially correlated measurements. It is therefore not surprising that active perception methods utilizing semantic measurements tend to use viewpoint-dependent models, which allow to better quantify information value of future views for planning. Atanasov et al. (2014) use a measurement model based on VP-Tree object detector Atanasov et al. (2014), parameterized by object class and orientation relative to camera, to perform non-myopic planning for inference of object class and orientation. Closely related to our method Velez et al. (2012) and Teacy et al. (2015) use Gaussian Processes (GPs) to model detector responses for objects of a given class, capturing spatial correlations by learning parameters of the GP kernels. Patten et al. (2018) also use per-class GP observation models of point-cloud features. However, the above assume robot localization is known, which in reality is a simplification. Becerra et al. (2016) relax the assumption of known localization, however, discretize pose space both for localizing the robot and in the viewpoint dependent class model representation.

As classifiers are ubiquitously based on Machine Learning and robots may be deployed in environments different from the ones the classifier had been trained on, the issue of covariate shift needs to be addressed. In recent years, several methods have been proposed for efficiently obtaining the (approximate) predictive uncertainty in the output of Deep Neural Networks associated with covariate shift, including MC-Dropout Gal and Ghahramani (2016), Bootstrapping Osband et al. (2016), ensembles Lakshminarayanan et al. (2017) and, recently, Prior Networks Malinin and Gales (2018). While these are used, among else, for efficient exploration and active learning Gal et al. (2017), Osband et al. (2016), safe decision making Lütjens et al. (2018), Malinin

et al. (2017), McAllister et al. (2017), semantic segmentation Kendall et al. (2015a), camera localization Kendall et al. (2015b), and the more closely related Miller et al. (2018a, b) who incorporate model uncertainty into merging strategy of object bounding box detections in a single frame, we are not aware of existing approaches for fusion of classification measurements over multiple frames/viewpoints attempting to account for covariate shift, except for the authors work Feldman and Indelman (2018a), Tchuiev and Indelman (2018), Tchuiev et al. (2019). Tchuiev and Indelman (2018) fuse class measurements carrying model uncertainty information represented with a Dirichlet distribution, however without considering localization inference or spatial correlation among measurements. Tchuiev et al. (2019) perform classification as part of hybrid inference under localization uncertainty, however not addressing correlations among viewpoints or model uncertainty.

While using most likely class measurements Bowman et al. (2017), Mu et al. (2016), Omidshafiei et al. (2016) provides a simple measurement model (categorical) and update rule for inference, it discards the response magnitude, which may be a useful indicator of the measurement quality. Likewise, using a single classifier response for the classified class Becerra et al. (2016), Teacy et al. (2015), i.e. a set of per-class detectors, discards information of correlation among responses, related to class aliasing. Eventually, particularly in the case of Deep Neural Networks, one can talk of a general semantic feature vector, of which classifier output is an example, which is a generalization of the above-mentioned measurements, as any can be computed out of it by choosing the maximal component or its index. A measurement model capturing variation of the entire feature vector thus has access to more information. For this reason, we model the entire classifier response rather than a single component. In the case that a new class needs to be added (offline), it suffices to create a new model describing responses of the same semantic feature detector for objects of the new class.

In the following section we formalize the problem, then proceed to describe our approach.

3 Notations and problem formulation

Consider a robot traversing an unknown environment, taking observations of different scenes. The robot's motion between times k and $k+1$ is initiated by a control input u_k, that may originate from a human user, or be determined by a motion planning algorithm. We denote the robot's pose at time instant k by x_k, and by $X_{0:k} = \{x_0, \ldots, x_k\}$ the sequence of poses up to that time. Let $\mathcal{H}_k = \{\mathcal{U}_{0:k-1}, \mathcal{Z}_{0:k}\}$ represent the history, comprising observations $\mathcal{Z}_{0:k} = \{z_0, \ldots, z_k\}$ and controls $\mathcal{U}_{0:k-1} = \{u_0, \ldots, u_{k-1}\}$ up until time k. We focus on the task of classification of a single object belonging to one of N_c known classes, denoted by indexes $\mathcal{C} = \{1, \ldots, N_c\}$.

Our goal is to maintain the classification posterior, or *belief*, at time instant k:

$$b[c_k] \doteq \mathbb{P}(c \mid \mathcal{H}_k). \tag{1}$$

The classification posterior is the probability of the object in question to belong to class $c \in \mathcal{C}$, given all measurements and user controls up to time k. In calculating this posterior we want to take into account spatial correlation among measurements, model uncertainty, as well as uncertainty in pose from which these measurements are taken (localization uncertainty).

3.1 Classifier measurements with uncertainty

In the Bayesian approach, given an observation z_k (in our context, an image) we are interested in obtaining the class (categorical) posterior

$$\mathbb{P}(s^{(i)} = c \mid z_k, \mathcal{D}), \tag{2}$$

where $s^{(i)}$ is the i'th component of a categorical vector s (i.e. $\sum_i s^{(i)} = 1$) and $\mathcal{D}$ is training data. For a given classifier unit, the vector θ of its parameters either determines its output as function of input, or otherwise can be regarded as a sufficient statistic. In either case, we can rewrite Eq. (2) as

$$\begin{aligned} &\mathbb{P}(s^{(i)} = c \mid z_k, \mathcal{D}) \\ &= \int_\theta \mathbb{P}(s^{(i)} = c \mid z_k, \theta) \cdot \underbrace{\mathbb{P}(\theta \mid \mathcal{D})}_{\substack{\text{Model} \\ \text{Uncertainty}}} d\theta. \end{aligned} \tag{3}$$

The second term above is referred to as *Model Uncertainty* or *Epistemic Uncertainty* Gal (2017), Kendall and Gal (2017), capturing uncertainty in classifier parameters given our training data. Uncertainty in θ induces a distribution over the values of $\mathbb{P}(s^{(i)} = c \mid z, \theta)$, a *distribution over distributions* Malinin and Gales (2018). The latter should intuitively be relatively concentrated near training data and spread out away from training data, reflecting the level of uncertainty in the classifier output.

We approximate the model uncertainty of the neural network classifier using MC-Dropout proposed by Gal and Ghahramani (2016), although this is not required by our approach and any other technique providing (approximate) samples from $\mathbb{P}(\theta \mid \mathcal{D})$ may be equally used. We chose this method for its simplicity, although Myshkov and Julier (2016) show that it may underestimate the model uncertainty in some cases. In short, for every classifier input we perform several forward passes applying random dropouts at each pass, to obtain a set of classification vectors, characterizing

the uncertainty. Figure 1b illustrates classifier measurements obtained over a track in the vicinity of an object of interest, when using MC-Dropout to approximate model uncertainty. Each measurement is a set of classification vectors (in the illustration—a point in the 3-d simplex corresponds to the case of 3 candidate classes) characterizing the model uncertainty, which generally varies among different viewpoints. Model uncertainty is complementary to the "classification confidence" which corresponds to the distance of the mean classification vector (red dot inside the simplex) from the corners.

Formally, we assume that the robot has at its disposal an object classifier unit, which, given observation z_k (e.g., an image), calculates a set of outputs $\mathcal{S}_k \triangleq \{s_k\}$, where each output $s_k \in \mathbb{R}^{N_c \times 1}$ represents a categorical belief over the class of the observed object, i.e. $\sum_{i=1}^{N_c} s_k^{(i)} = 1$.

The set S_k can be interpreted as an approximation to the predictive distribution from Eq. (2),

$$\forall s \in \mathcal{S}_k \;\; s \sim \mathbb{P}(\cdot \mid z_k, \mathcal{D}), \tag{4}$$

carrying information of the classifier's model uncertainty for the given input z_k.

3.2 Viewpoint-dependent class model

For the class likelihood we use a model similar to the one proposed by Teacy et al. Teacy et al. (2015). For a single classifier measurement s (categorical vector) captured from relative pose $x^{(rel)}$, the class likelihood is a probabilistic model

$$\mathbb{P}(s \mid c, x_k^{(rel)}), \tag{5}$$

where $c \in \mathcal{C}$ is the object class, and the k subscript denotes time index. Denoting object pose in global frame as o we can explicitly write

$$x_k^{(rel)} \doteq x_k \ominus o. \tag{6}$$

The dependence of the model in Eq. (5) on viewpoint naturally captures view-dependent variations in object appearance and as a result in classifier responses as illustrated in Fig. 1a. Further, to incorporate the notion that similar views tend to yield similar classifier responses and in particular, are not independent, we consider the joint distribution

$$\mathbb{P}(\mathcal{S}_{0:k} \mid c, \mathcal{X}_{0:k}^{(rel)}), \tag{7}$$

characterizing the classification unit's outputs $\mathcal{S}_{0:k} \doteq \{S_0, \ldots, S_k\}$ when viewing an object of class c from a sequence of relative poses $\mathcal{X}_{0:k}^{(rel)} \doteq \{x_0^{(rel)}, \ldots, x_k^{(rel)}\}$. Similar to Teacy et al. (2015), Velez et al. (2012), we represent this joint distribution with a Gaussian Process, learned using the classifier unit. Explicitly, we model training set classifier response when viewing object of class c from relative pose $x^{(rel)}$ as

$$s^{(i)} = f_{i|c}(x^{(rel)}) + \varepsilon, \tag{8}$$

where the i index denotes component i of classification vector s, $\varepsilon \sim N(0, \sigma_n^2)$ i.i.d. noise, and (dropping the (rel) superscript for clarity)

$$f_{i|c}(x) \sim \mathcal{GP}\left(\mu_{i|c}(x), k_{i|c}(x, x)\right), \tag{9}$$

where $\mu_{i|c}$ and $k_{i|c}$ are the mean and covariance functions defining the GP

$$\mu_{i|c}(x) = \mathbb{E}\{s^{(i)} \mid c, x\} \tag{10}$$

$$k_{i|c}(x, x') = \mathbb{E}\{(f_{i|c}(x) - \mu_{i|c}(x))(f_{i|c}(x') - \mu_{i|c}(x'))\} \tag{11}$$

We thus model the classification vector for each class c with independent, per-component GP's. Note also the Gaussian approximation of the distribution of the classification vector, which resides in the simplex (other representations exist, which however are not readily interpreted as a spatial model).

For the covariance we use the squared exponential function:

$$k_{i|c}(x, x') = \sigma_{i|c}^2 \exp\left(-\frac{1}{2}(x - x')^T L_{i|c}^{-1}(x - x')\right), \tag{12}$$

where $\sigma_{i|c}^2$ is the variance, and $L_{i|c}$ is the length scale matrix (of the dimension of x), determining the rate of the covariance decay with distance. These parameters can be learned from training data Rasmussen and Williams (2006).

Denote the training set for class c as $\{S_T^c, X_T^c\}$, with S_T^c classifier measurements, and X_T^c the corresponding poses, and denote (test-time) measurements as $S = \mathcal{S}_{0:k}$ and $X = \mathcal{X}_{0:k}^{(rel)}$. Note that training set classifier measurements are obtained without dropout i.e. they do not carry model uncertainty information. Further, the following equations Eqs. (13–16) all hold per vector-component (joined in Eq. (17)), i.e. for simplifying notation we drop the i index in $\mathcal{S}^{(i)}$, $\mathcal{S}_T^{(i)}$ and $k_{i|c}$.

We follow Rasmussen and Williams (2006) and model the joint distribution of classifier measurements (per-component) for object of class c as

$$\mathbb{P}(S_T^c, S \mid c, X_T^c, X) = N\left(0, \begin{bmatrix} K_c(X_T^c, X_T^c) + \sigma_n^2 I & K_c(X_T^c, X) \\ K_c(X, X_T^c) & K_c(X, X) \end{bmatrix}\right), \tag{13}$$

where S_T^c are the classifier measurements used for training, not carying model uncertainty information, $S = \mathcal{S}_{0:k}$ are the

test-time measurements, such that $\mathcal{S}_j \sim \mathbb{P}(\cdot \mid z_j)$ as in Eq. (4), and K_c is the matrix produced by application of kernel k_c on all pairs of input vectors. We thus obtain the conditional distribution for classifier measurements of object of class c

$$\mathbb{P}(\mathcal{S}_{0:k} \mid c, X_T^c, S_T^c, \mathcal{X}_{0:k}^{(rel)}) = N(\mu, \Sigma), \tag{14}$$

with

$$\mu = K_c(X, X_T^c) \cdot H \cdot S \tag{15}$$
$$\Sigma = K_c(X, X) - K_c(X, X_T^c) \cdot H \cdot K_c(X_T^c, X), \tag{16}$$

and where $H \doteq \left(K_c(X_T^c, X_T^c) + \sigma_n^2 I\right)^{-1}$.

Going back to the time notation, we finalize by combining the per-component models into a joint class likelihood as

$$\mathbb{P}(\mathcal{S}_{0:k} \mid c, X_T^c, S_T^c, \mathcal{X}_{0:k}^{(rel)}) = \prod_i \mathbb{P}(\mathcal{S}_{0:k}^{(i)} \mid c, X_T^c, S_T^{c,(i)}, \mathcal{X}_{0:k}^{(rel)}), \tag{17}$$

where the superscript (i) refers to vector component, as above. Note that this formulation somewhat differs from Teacy et al. (2015), where inference from training data is done by offline learning of GP mean and covariance functions rather than using a joint distribution as in Eq. (13). Also note that while for simplicity we defined the model in Eq. (13) per-component (considering components as independent), subsequent development also holds for a joint model, i.e. jointly referring to all components of classifier measurements in Eqs. (13–16), with full covariance in Eq. (14) rather than block-diagonal as induced by Eq. (17).

3.3 Assumptions

We briefly review basic assumptions in the context of the formulation above. We assume that we have access to a **Bayesian classifier**, which given raw measurement z provides samples from a probability distribution over classification vectors characterizing the predictive uncertainty $\mathbb{P}(s \mid z, \mathcal{D})$, see Eq. (4). We assume that **viewpoint-dependent class models** are available, modeling the spatial responses $\mathcal{S}_{0:k}$ of that classifier given poses relative to object $\mathcal{X}_{0:k}^{(rel)}$, in a joint Gaussian distribution $\mathbb{P}(\mathcal{S}_{0:k} \mid c, \mathcal{X}_{0:k}^{(rel)})$ for objects of class c, where the Gaussian form is required for closed-form calculations in the following. As described in the previous section, Gaussian Process regressors can be learned to implement such models. Existence of pre-trained models for known classes implies **closed set** conditions, i.e. the possible object classes are known in advance. Objects of novel classes are assumed to trigger high predictive uncertainty in the classifier output (in itself an assumption, since predictive uncertainty depends on the classifier training set), reflected in uncertainty in the posterior. Further, the formulation focuses on classification of a single, static object. In general scenes comprising multiple objects this formulation can be applied for each object separately, under the assumption of **known data association**, and the additional underlying assumption of negligibility of the effect of occlusions. Finally, in this work we assume that the **orientation** of the object relative to the robot **is known** (perhaps, detected from observations). The need for this assumption becomes apparent and is briefly discussed in Sect. 4.2. Under these assumptions, in the following section we detail our approach to the classification problem formulated above.

4 Approach

To account for both localization and model uncertainty we rewrite Eq. (1) as marginalization over latent robot and object poses, and over classifier outputs. We start by marginalizing over robot pose history and object pose

$$b[c_k] = \mathbb{P}(c \mid \mathcal{H}_k) = \int_{\mathcal{X}_{0:k}, o} \mathbb{P}(c, \mathcal{X}_{0:k}, o \mid \mathcal{H}_k)\, d\mathcal{X}_{0:k} do, \tag{18}$$

which, using chain rule, can be written as

$$b[c_k] = \int_{\mathcal{X}_{0:k}, o} \underbrace{\mathbb{P}(c \mid \mathcal{X}_{0:k}, o, \mathcal{H}_k)}_{(a)} \underbrace{\mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k)}_{(b)}\, d\mathcal{X}_{0:k} do. \tag{19}$$

Term (a) above is the classification belief given relative poses $\mathcal{X}_{0:k}^{(rel)}$ which are calculated from $\mathcal{X}_{0:k}$ and o via Eq. (6). Term (b) represents the posterior over $\mathcal{X}_{0:k}$ and o given observations and controls thus far. As such, this term can be obtained from existing SLAM approaches. One can further rewrite the above equation as

$$b[c_k] = \mathop{\mathbb{E}}_{\mathcal{X}_{0:k}, o} \{\mathbb{P}(c \mid \mathcal{X}_{0:k}^{(rel)}, \mathcal{H}_k)\}, \tag{20}$$

where the expectation is taken with respect to the posterior $p(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k)$ from term (b). In practice, this posterior can be computed using SLAM methods (see Section 4.2), which commonly model it with a Gaussian distribution. We then use the obtained distribution to approximate the expectation in Eq. (20) using sampling.

In the following we detail the computation of the terms (a) and (b) of Eq. (19).

4.1 Classification under known localization

In this section we develop the update of classification belief given known pose history, term (a) in Eq. (19), when receiving new measurements at time step k, while accounting for correlations with previous measurements and model uncertainty.

To simplify notation, we shall denote history of observations, controls and (known) relative poses as

$$H_k \doteq \mathcal{H}_k \cup \mathcal{X}_{0:k}^{(rel)} \equiv \{\mathcal{U}_{0:k-1}, \mathcal{Z}_{0:k}, \mathcal{X}_{0:k}^{(rel)}\}. \tag{21}$$

We start by marginalizing term (a) over model uncertainty in the classifier output at time k

$$\mathbb{P}(c \mid H_k) = \int_{s_k} \mathbb{P}(c \mid s_k, H_k) \cdot \mathbb{P}(s_k \mid H_k)\, ds_k. \tag{22}$$

Assuming s_k carries the class information from measurement z_k, and that $s_k \sim \mathbb{P}(s_k \mid z_k)$ we can rewrite this as

$$\mathbb{P}(c \mid H_k) = \int_{s_k} \mathbb{P}(c \mid s_k, H_k \setminus \{z_k\}) \cdot \mathbb{P}(s_k \mid z_k)\, ds_k. \tag{23}$$

In our case, $\{s_k\}$ are samples from $\mathbb{P}(s_k \mid z_k)$, so we can approximate the integral as

$$\mathbb{P}(c \mid H_k) \approx \frac{1}{n_k} \sum_{s_k \in \mathcal{S}_k} \mathbb{P}(c \mid s_k, H_k \setminus \{z_k\}). \tag{24}$$

To calculate the summand, we apply Bayes' law and then smoothing over class in the denominator

$$\begin{aligned} &\mathbb{P}(c \mid H_k) \\ &= \sum_{s_k} \frac{\eta(s_k)}{n_k} \cdot \mathbb{P}(s_k \mid c, H_k \setminus \{z_k\}) \cdot \mathbb{P}(c \mid H_k \setminus \{z_k\}) \end{aligned} \tag{25}$$

with

$$\eta(s_k) \doteq 1 / \sum_{c \in \mathcal{C}} \mathbb{P}(s_k \mid c, H_k \setminus \{z_k\}) \mathbb{P}(c \mid H_k \setminus \{z_k\}). \tag{26}$$

Note that the denominator in $\eta(s_k)$ is a sum of numerator (summand) terms in Eq. (25) for the different classes and can be computed efficiently (but cannot be discarded altogether due to the dependence on s_k). Further, note that

$$\begin{aligned} &\mathbb{P}(c \mid H_k \setminus \{z_k\}) = \mathbb{P}(c \mid \mathcal{X}_{0:k}^{(rel)}, \mathcal{Z}_{0:k-1}) \\ &= \mathbb{P}(c \mid \mathcal{X}_{0:k-1}^{(rel)}, \mathcal{Z}_{0:k-1}) = \mathbb{P}(c \mid H_{k-1}). \end{aligned} \tag{27}$$

As $\mathbb{P}(c \mid H_{k-1})$ has been computed in the previous step, we are left to compute the class likelihood term $\mathbb{P}(s_k \mid c, H_k \setminus \{z_k\})$. This term involves past observations $\mathcal{Z}_{0:k-1}$ but not classifier outputs $\mathcal{S}_{0:k-1}$, which need to be introduced to account for spatial correlation with s_k using Eq. (7). Marginalizing over $\mathcal{S}_{0:k-1}$ (recall that in our notation $\mathcal{S}_{0:k-1} \cup \{s_k\} = \mathcal{S}_{0:k}$) yields

$$\begin{aligned} &\mathbb{P}(s_k \mid c, H_k \setminus \{z_k\}) = \int_{\mathcal{S}_{0:k-1}} \mathbb{P}(\mathcal{S}_{0:k} \mid c, H_k \setminus \{z_k\})\, d\mathcal{S}_{0:k-1} \\ &= \int_{\mathcal{S}_{0:k-1}} \mathbb{P}(s_k \mid c, \mathcal{S}_{0:k-1}, H_k \setminus \{z_k\}) \\ &\quad \cdot \mathbb{P}(\mathcal{S}_{0:k-1} \mid c, H_k \setminus \{z_k\})\, d\mathcal{S}_{0:k-1}, \end{aligned} \tag{28}$$

where we applied smoothing to separate between past classifier outputs $\mathcal{S}_{0:k-1}$ for which observations $\mathcal{Z}_{0:k-1}$ are given and the current output s_k. The first term in the product reduces to $\mathbb{P}(s_k \mid c, \mathcal{S}_{0:k-1}, \mathcal{X}_{0:k}^{(rel)})$, a conditioned form of the class model Eq. (14) (and thus Gaussian, which we treat explicitly later in Eq. (31) and on). This term represents the probability to obtain classification s_k when observing an object of class c from relative pose $x_k^{(rel)}$ given previous classification results and relative poses. The second term in Eq. (28) can be approximated using Eq. (4) for the individual observations z_i, i.e.

$$\mathbb{P}(\mathcal{S}_{0:k-1} \mid c, H_k \setminus \{z_k\}) = \mathbb{P}(\mathcal{S}_{0:k-1} \mid \mathcal{Z}_{0:k-1}) \approx \prod_{i=0}^{k-1} \mathbb{P}(s_i \mid z_i)$$

Note that class c and poses $\mathcal{X}_{0:k-1}^{(rel)}$, both members of H_k can be omitted since conditioning on observations determines classifier outputs up to uncertainty due to classifier intrinsics (model uncertainty). The approximation is in the last equality, since in general classifier outputs $s_0, \ldots, s_{k-1}$ are interdependent through classifier parameters. We can now rewrite $\mathbb{P}(s_k \mid c, H_k \setminus \{z_k\})$ from Eq. (28) as

$$\int_{\mathcal{S}_{0:k-1}} \mathbb{P}(s_k \mid c, \mathcal{S}_{0:k-1}, \mathcal{X}_{0:k}^{(rel)}) \cdot \prod_{i=0}^{k-1} \mathbb{P}(s_i \mid z_i)\, d\mathcal{S}_{0:k-1}. \tag{29}$$

Assuming classifier output Eq. (4) is Gaussian, we denote

$$\mathbb{P}(s_i \mid z_i) = N(\mu_{z_i}, \Sigma_{z_i}), \tag{30}$$

where μ_{z_i} and Σ_{z_i} are estimated from $\mathcal{S}_i$. Since class model is Gaussian, see Eq. (14), the first term in the integrand in Eq. (29) is a Gaussian that we denote as

$$\mathbb{P}(s_k \mid c, \mathcal{S}_{0:k-1}, \mathcal{X}_{0:k}^{(rel)}) = N(\mu_{k|0:k-1}, \Sigma_{k|0:k-1}) \tag{31}$$

where, utilizing standard Gaussian Process equations Rasmussen and Williams (2006),

$$\mu_{k|0:k-1} = \mu_k + \Omega \cdot (\mathcal{S}_{0:k-1} - \mu_{0:k-1}) \tag{32}$$

$$\Sigma_{k|0:k-1} = K(x_k, x_k) - \Omega \cdot K(\mathcal{X}_{0:k-1}, x_k) \tag{33}$$

with $\Omega \doteq K(x_k, \mathcal{X}_{0:k-1}) K(\mathcal{X}_{0:k-1}, \mathcal{X}_{0:k-1})^{-1}$.

Using these relations, the integrand from Eq. (29) is a Gaussian distribution over $\mathcal{S}_{0:k}$, that can be inferred as follows.

$$\begin{aligned}&\mathbb{P}(s_k \mid c, \mathcal{S}_{0:k-1}, \mathcal{X}_{0:k}^{(rel)}) \cdot \prod_{i=0}^{k-1} \mathbb{P}(s_i \mid z_i) \\ &= \eta \exp\left\{-\frac{1}{2}\left(\|s_k - \mu_{k|0:k-1}\|^2_{\Sigma_{k|0:k-1}} + \sum_{i=0}^{k-1} \|s_i - \mu_{z_i}\|^2_{\Sigma_{z_i}}\right)\right\},\end{aligned} \tag{34}$$

where η only depends on $\mathcal{X}_{0:k}^{(rel)}$. Using Eq. (32) we can write

$$\begin{aligned}s_k - \mu_{k|0:k-1} &= s_k - \mu_k - \Omega \cdot (\mathcal{S}_{0:k-1} - \mu_{0:k-1}) \\ &= [-\Omega \quad I]\,(\mathcal{S}_{0:k} - \mu_{0:k})\end{aligned} \tag{35}$$

We have that the integrand Eq. (34) from Eq. (29) is proportional to a joint Gaussian distribution $N(\mu_J, \Sigma_J)$ with

$$\Sigma_J = \left(\Sigma_s^{-1} + \Sigma_z^{-1}\right)^{-1} \tag{36}$$

$$\mu_J = \Sigma_J \cdot \left(\Sigma_s^{-1}\mu_s + \Sigma_z^{-1}\mu_z\right), \tag{37}$$

where

$$\mu_s = \begin{pmatrix}\mu_0 \\ \vdots \\ \mu_{k-1} \\ \mu_k\end{pmatrix} \quad \mu_z = \begin{pmatrix}\mu_{z_0} \\ \vdots \\ \mu_{z_{k-1}} \\ 0\end{pmatrix}, \tag{38}$$

and

$$\Sigma_s^{-1} = [-\Omega \quad I]^T\, \Sigma_{k|0:k-1}^{-1}\, [-\Omega \quad I] \tag{39}$$

$$\Sigma_z^{-1} = \begin{pmatrix}\Sigma_{z_0}^{-1} & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \Sigma_{z_{k-1}}^{-1} & 0 \\ 0 & \cdots & 0 & 0\end{pmatrix} \tag{40}$$

Finally, the class likelihood from Eq. (28) is the marginal distribution of the above. Specifically, the integral is directly calculated by evaluation at s_k of a Gaussian PDF with the components corresponding to s_k from μ_J and Σ_J as mean and covariance.

So far, we have described how to update the class belief given known localization, term (a) of Eq. (19), upon arrival of new measurements. We now proceed to describe how the localization belief, term (b), is computed.

4.2 Localization inference

In this work we assume that the object orientation relative to the robot is known (leaving o with 3 degrees of freedom), and so term (b) of Eq. (19) is essentially a SLAM problem with robot pose history $\mathcal{X}_{0:k}$ and one landmark, the target object localization o, to be inferred. Specifically, we can express the target distribution as marginalization over all landmarks $\mathcal{L}$, except for the object of interest

$$\mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k) = \int_{\mathcal{L}} \mathbb{P}(\mathcal{X}_{0:k}, o, \mathcal{L} \mid \mathcal{H}_k)\, d\mathcal{L}. \tag{41}$$

This can be computed using state of the art methods, e.g. if a Gaussian posterior is assumed, using an efficient solver such as iSAM2 Kaess et al. (2012). Note that the above assumption of Gaussian posterior is not a requirement of the formulation in Eq. (19)—indeed, using sampling allows the localization posterior to be of any distribution provided it can be sampled from.

Also note that the history $\mathcal{H}_k$ contains both geometric and semantic measurements, allowing related work to leverage the viewpoint-dependent model in inference as a semantic-geometric factor Kopitkov and Indelman (2018), Tchuiev et al. (2019). Our case is slightly different however, as the formulation accounts for model uncertainty and correlation among viewpoints. In this work we limit ourselves to usage of semantic factors for semantic (term (a) of Eq. (19)) but not geometric inference (geometry helps semantics Cadena et al. (2016)), requiring relative object orientation to be externally known.

4.3 Overall algorithm

Algorithm 1 summarizes computations for time step k in the localization and model uncertainty aware classification approach described above. Observations $\mathcal{Z}_k$ are passed through the classifier unit to obtain semantic measurements and model uncertainty, at which point summary statistics can be computed, for use in Eq. (29). To account for localization uncertainty, robot and object poses are sampled from the current (updated) posterior. For each sample, class model predictions (mean and covariance matrices) are computed for all classes. To account for model uncertainty, measurement likelihood is computed for each $s_k^{(i)} \in \mathcal{S}_k$, then averaged. Note that formulation using the conditional distribution of s_k given past measurements in Eq. (29) allows efficient marginalization in Eq. (25) and computation of normalizing constant $\eta(s_k)$ in Eq. (26), which would otherwise require intractable summation over combinations of individual semantic measurements. Similarly, using summary statistics (fitting a Gaussian per time-step) over past semantic measurements allows the marginalization in Eq. (29) to be

Algorithm 1 Localization and model uncertainty aware classification.

This describes computations for time step k.

Input: Localization posterior $\mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k)$, new observations $\mathcal{Z}_k$

1: $\mathcal{S}_k = \{s_k^{(i)}\}_{i=1}^{n_k} \leftarrow$ ClassifyWithDropout($\mathcal{Z}_k$), pre-calculate Σ_z^{-1}, μ_z (Eq. (40), Eq. (38))
2: Sample $\left\{\mathcal{X}_{0:k}^{(i)}, o^{(i)}\right\}_{i=1}^{n_x} \sim \mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k)$
3: **for** $\mathcal{X}_{0:k}, o \in \{\mathcal{X}_{0:k}, o\}$ **do**
4: $\quad \forall c \in \mathcal{C}$ calculate Σ_J, μ_J (Eq. (36), Eq. (37)) ▷ These are per-class through GP model
5: $\quad$ **for** $s_k \in \mathcal{S}_k$ **do**
6: $\quad\quad$ **for** $c \in \mathcal{C}$ **do**
7: $\quad\quad\quad \mathbb{P}(s_k \mid c, H_k \setminus \{\mathcal{Z}_k\}) \leftarrow N(s_k; \mu_J^{(k)}, \Sigma_J^{(k,k)})$ ▷ Likelihood given past observations Eq. (29)
8: $\quad\quad\quad \tilde{h}^c \leftarrow \mathbb{P}(s_k \mid c, H_k \setminus \{\mathcal{Z}_k\}) \cdot \mathbb{P}(c \mid H_{k-1})$ ▷ Unnormalized class likelihood, Eq. (25)
9: $\quad\quad$ **end for**
10: $\quad\quad \mathbb{P}(c \mid s_k, H_k) \leftarrow \tilde{h}^c / \eta(s_k), \quad \eta(s_k) = \sum_c \tilde{h}^c$ ▷ Normalize class likelihood, Eq. (25)
11: $\quad$ **end for**
12: $\quad \mathbb{P}(c \mid H_k)^{(i)} \leftarrow \sum_{s_k} \mathbb{P}(c \mid s_k, H_k)/n_k$ ▷ Classification given localization Eq. (24)
13: **end for**
14: $\mathbb{P}(c \mid \mathcal{H}_k) \leftarrow \sum_i \mathbb{P}(c \mid H_k)^{(i)}/n_x$

Note: Lines 10, 12, 14 apply per-class, $\forall c \in \mathcal{C}$. In step 7, superscripts select elements corresponding to time k in the joint mean vector and covariance matrix.

computed in closed form. Classifications thus computed per localization sample are averaged to yield the final output. In the next section we analyze the computational complexity of Algorithm 1, then present the experimental evaluation and results.

4.4 Computational aspects

While using samples allows in principle to approximate arbitrary non-tractable or unknown probability distributions, on the downside it involves computations repeated per-sample, inducing a sharp precision vs. runtime tradeoff on the number of samples, especially so as state dimension grows over time. Additionally, in classification, most computations are repeated per class. Finally, complexity depends on the number of time steps k, which affects both the overall number of observations to be processed, and the number of state variables in the smoothing formulation. Note that k value relevant to the classification of an object is the number of time steps it has been observed, as other poses can be dropped from the marginalization in Eq. (19), since measurement model and thus classification is conditionally independent of them.

In this section we analyze the theoretical complexity of the proposed method, then present an approach for incremental sampling of robot and object poses allowing to reduce repeated calculations, and discussion.

4.4.1 Theoretical complexity

We denote the number of model uncertainty samples (per time step) in step 1 of Algorithm 1 as N_s, number of localization samples in step 2 as N_x and number of classes as N_c, and use them to express complexity. We assume these are constant throughout the run. We further denote as N_f the dimension of the semantic feature vector, i.e. $\forall k, i\ s_k^{(i)} \in \mathbb{R}^{N_f}$. In our implementation, $N_f \equiv N_c$ as we use the raw classification vectors as features.

Step 1 depends on the classification and model uncertainty computation method. In the case of a Neural Network classifier with a fixed architecture using MC-dropout, we assume each forward pass to take constant time, making complexity linear in the number of forward passes $O(N_s)$. In the same step, computation of μ_z takes $O(N_f N_s)$ and of Σ_z^{-1} takes $O(N_s N_f^2 + N_f^3)$, which is also the overall complexity for this step. Step 2 involves sampling an entire trajectory (MCMC) at each time step i.e. $O(k \cdot N_x)$. Step 4 involves (1) prediction of μ_s (2) computation of $\Sigma_{k|0:k-1}$ and of Σ_s^{-1} (see Eq. (33) and Eq. (39)), and (3) Computations of Σ_J and μ_J (see Eq. (36) and Eq. (37)). These values depend on the robot trajectory sampled in step 2 and on the class models. Complexity is dominated by the inversion in Eq. (36) at $O(N_f k^3)$ when done separately for each semantic feature. Calculations in this step can be reduced if trajectory samples from previous time steps are re-used, as discussed in Sect. 4.4.2. In all, complexity for step 4 is $O(N_c N_f k^3)$ as computation is done for each class. Since Σ_J, μ_J are precomputed in step 4, calculation of likelihood in step 7 takes $O(N_f)$ (covariance-form marginalization, followed by evaluation of 1-dimensional Gaussian PDFs for N_f features). Step 8 incurs no additional cost (i.e., $O(1)$) when $\mathbb{P}(c \mid H_{k-1})$ can be assumed pre-computed, which corresponds to re-use of localization samples from previous time indexes (see Sect. 4.4.2). In this case complexity of inner loop steps 5–12 is $O(k^3 N_c N_s N_f)$. Otherwise, if past localization samples are not re-used, computation from scratch is required, which can be done by repeating steps 4–12 to incrementally compute $\mathbb{P}(c \mid H_1), \mathbb{P}(c \mid H_2), \ldots, \mathbb{P}(c \mid H_{k-1})$, which also requires that semantic measurements from all previous time steps be kept. Incremental computation for time indexes $0, \ldots, k$ results in complexity $O(k^4 N_c N_s N_f)$. Overall, total time complexity for time step k stands at $O(k^4 N_c N_s N_f N_x + N_s N_f^2 + N_f^3)$ if class posterior in step 8 needs to be recomputed, or $O(k^3 N_c N_s N_f N_x + N_s N_f^2 + N_f^3)$ if it does not, where k is the number of viewpoints from which the object was observed.

4.4.2 Incremental sampling

Step 2 in Algorithm 1 involves sampling of robot trajectory, a state vector which grows over time, in order to compute the marginalization in Eq. (19), accounting for localization uncertainty. This marginalization must involve a history of robot poses because of the measurement model (GP), Eq. (7), which captures correlations among present and past views.

While, as noted in Sect. 4.4, this sampling can be limited to poses at time indices corresponding to views of the object, a further runtime improvement can be achieved if trajectory samples from past time indices can be re-used in samples from the updated posterior.

Specifically, sample reuse would benefit step 4, where some values can be updated instead of being re-calculated, and especially step 8 where values from past time steps can be re-used eliminating the required calculations to $O(1)$, in addition to directly reducing sampling in step 2.

Formally, assuming at time step k we can use samples $\{\mathcal{X}_{0:k}^{(i)}, o^{(i)}\}_{i=1}^{N_s} \sim \mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k)$ to approximate integral Eq. (19) with sum

$$\begin{aligned}\mathbb{P}(c \mid \mathcal{H}_k) &= \int_{\mathcal{X}_{0:k},o} \mathbb{P}(c \mid \mathcal{X}_{0:k}, o, \mathcal{H}_k) \\ &\cdot \mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k)\, d\mathcal{X}_{0:k} do \\ &\approx \sum_i \mathbb{P}(c \mid \mathcal{X}_{0:k}^{(i)}, o^{(i)}, \mathcal{H}_k) \\ &\cdot \mathbb{P}(\mathcal{X}_{0:k}^{(i)}, o^{(i)} \mid \mathcal{H}_k)\, d\mathcal{X}_{0:k} do, \end{aligned} \tag{42}$$

at time step $k+1$ we would like to approximate

$$\begin{aligned}\mathbb{P}(\mathcal{X}_{0:k+1}, o \mid \mathcal{H}_{k+1}) &= \mathbb{P}(x_{k+1} \mid \mathcal{X}_{0:k}, o \mid \mathcal{H}_{k+1}) \\ &\cdot \mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_{k+1}) \end{aligned} \tag{43}$$

without sampling the variables from scratch, ideally keeping samples of $\mathcal{X}_{0:k}, o$ from time step k, and only sampling incrementally $x_{k+1}^{(i)} \sim \mathbb{P}(\cdot \mid \mathcal{X}_{0:k}^{(i)}, o^{(i)}, \mathcal{H}_{k+1})$. Since generally $\mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_{k+1}) \neq \mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k)$ a possible approach to sample reuse is to weight incremental samples of trajectory $\mathcal{X}_{0:k+1}$ by the ratio $\mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_k)/\mathbb{P}(\mathcal{X}_{0:k}, o \mid \mathcal{H}_{k+1})$ when approximating the integral from Eq. (42) at step $k+1$, an idea known as importance sampling. Under this approach, full posterior samples only need to be generated to replace incremental samples with low weights, allowing reuse of samples and calculations, see, e.g., Farhi and Indelman (2019).

4.4.3 Discussion (computational aspects)

The method as presented in the previous section allows efficient computation with respect to model uncertainty offering a recursive formulation for model uncertainty samples, with marginalization at the current time step approximated with samples Eq. (24), and past time steps marginalized out analytically Eq. (40). However, formulation for localization uncertainty is not recursive due to the batch sampling of entire robot trajectory and object pose in Eq. (20) required there to account for correlations with past time steps. In the previous clause Sect. 4.4.2 we mentioned a possible approach to allow recursive computation with respect to localization samples as well, rendering the method fully incremental, which could be addressed in future research. In practice (and even for fully incremental computations) there is a tradeoff between runtime and accurate treatment of uncertainty. One approach is to not address uncertainty at all, equivalent to using a single sample (e.g. max-likelihood). Another possibility is to neglect correlations to some past time steps. Both can be considered heuristics in the general scheme described above.

5 Results

We evaluate our method in a MATLAB simulation using synthetically generated classifier measurements and in a simulated 3D environment using Unreal Engine (UE) with UnrealCV Qiu et al. (2017), with semantic measurements obtained from a CaffeNet classifier Jia et al. (2014). In both settings, we split our evaluation into scenarios of model uncertainty and of localization uncertainty, comparing performance to baseline methods, as detailed in the next section. To validate our contributions in a realistic setting, we additionally present an evaluation using a real-world dataset, Active Vision Dataset Ammirato et al. (2017), using AlexNet Krizhevsky et al. (2012) for semantic measurements.

In all settings, GP models (using Scikit-Learn Pedregosa et al. (2011)) are fit offline for each of the candidate classes. Following Eq. (17), components of the classification vector output by the classifier are considered independent, amounting to a separate GP per component. Another simplifying assumption is that object orientation relative to robot is known, following Sect. 3.3.

5.1 Compared approaches and performance metrics

We compare the results of three methods. Ours we denote `Model with Uncertainty`, which takes into account spatial correlations, as well as uncertainty in pose and classifier model uncertainty. The second is `Model Based`, similar to the method described by Teacy et al. Teacy et al. (2015) but with GP defined as in Eq. (13) (and Rasmussen and Williams (2006)), which takes into account spatial correlation, but not localization nor model uncertainty The third is `Simple Bayes`, which directly uses the classifier scores and assumes spatial independence between observations, as in e.g. Patten et al. Patten et al. (2016).

We compare the methods above with relation to the following metrics: (i) probability of ground-truth class; (ii) mean squared detection error; and (iii) most-likely-to-ground-truth ratio. The mean squared detection error (MSDE) is defined as

$$MSDE \doteq \frac{1}{N_c} \sum_{c' \in \mathcal{C}} \left(\mathbb{1}_c(c') - \mathbb{P}(c' \mid \mathcal{H}) \right)^2 \tag{44}$$

here c is the ground truth class and $\mathbb{1}_c(c')$ is 1 if $c = c'$ and 0 otherwise. This measure was also used in Teacy et al. (2015).

The most-likely-to-ground-truth ratio (MGR) is defined as

$$MGR \doteq \frac{\text{argmax}_{c'} \ \mathbb{P}(c' \mid \mathcal{H})}{\mathbb{P}(c \mid \mathcal{H})} \tag{45}$$

for ground truth class c. Roughly, this measure penalizes high confidence in the wrong class. In a way it "demands" ground truth class to be most (possibly, equally) likely. Finally, as a sanity check, we also present values of the "correct ratio" (CR), defined as

$$\begin{aligned} CR &\doteq \mathbb{E}_{\mathcal{H}} \left\{ \mathbb{1}_c \left(\text{argmax}_{c'} \, \mathbb{P}(c' \mid \mathcal{H}) \right) \right\} \\ &\approx \frac{1}{N} \sum_{\mathcal{H}} \mathbb{1}_c \left(\text{argmax}_{c'} \, \mathbb{P}(c' \mid \mathcal{H}) \right), \end{aligned} \tag{46}$$

with $\mathbb{1}$ the indicator function as above, N the number of averaged terms and with the proposal distribution of $\mathcal{H}$ defined according to reported context. This defines the CR as the ratio of correct classifications, considering a classification result vector "correct" if it assigns the highest probability to the ground truth class, i.e. if the ground truth class is the most likely. Below we report *final CR*, i.e. where $\mathcal{H}$ includes all measurements from the entire trajectory, and *overall CR*, where we also average over measurement history at intermediate time steps along trajectories, to capture the method's behavior over time.

We now proceed to detail the experiments and the results.

5.2 MATLAB simulation results

We present experimental results for a MATLAB simulation, in which synthetic classifier measurements are generated using the GP model of the ground truth class, along a pre-determined track. Synthetic measurements are generated according to the procedure detailed in Algorithm 2. The class inference algorithm needs to fuse these measurements into a posterior over classes, essentially identifying which of the known GP models is the more likely origin of the measurements. We study robustness of our algorithm to model and localization uncertainty, and compare it to the state of the art.

5.3 Simulation experiments

Statistics for the three algorithms have been collected for several scenarios over realizations of simulated classification. In each scenario, GP models were created for three classes, by manually specifying the classifier response for chosen relative locations around the origin (i.e. locations assumed to be in object-centered coordinates) in the 2D plane, see Fig. 1a. Note that GP model for a class describes classifier responses for *all* classes, (see Eq. (17) and Sect. 3.2).

During simulation, the robot moves along a pre-specified trajectory and observes a single object from different viewpoints, see Fig. 1b for an example trajectory. At each time step the algorithm receives new classifier measurements and updated pose belief (simulating localization obtained from a SLAM solution). Classifier measurements are generated using the GP model of a "ground truth" class (the simulation of measurements is detailed in the next subsections), which needs to be inferred by the algorithm using the measurements.

We next present results on two scenarios highlighting our main contributions.

5.3.1 Model uncertainty scenario

Model uncertainty expresses the reliability of the classifier output. High model uncertainty corresponds to situations where classifier input that is far from the training data, often due to an unfamiliar scene, object or viewpoint pictured, causes output that may be arbitrary. We simulate this with two steps, performed at each time instant: first, a nominal "true" measurement $s^{nominal}$ is generated from a GP model of the ground truth class. The level of model uncertainty σ_u^2 is selected at each time step uniformly between 0 and σ_{max}^2 (a parameter). It is then used as standard deviation of a Gaussian centered at the true measurement to generate a simulated noised measurement s^{noised}. The `Model Based` and `Simple Bayes` algorithms receive s^{noised} as classification measurement and are not aware of the uncertainty. Our method receives samples (simulating outputs of several forward passes applying dropouts) drawn from a Gaussian distribution centered at s^{noised} with standard deviation σ_u^2. Algorithm 2 summarizes this process.

First scenario shows the effects of considerable model uncertainty, with no localization errors (perfect localization). Figure 2 shows plots of GP model of ground truth class and simulated classifier measurements (s^{noised}) over robot track (left) and per-component as a function of time (right). Figure 3 shows the statistics described above (probability assigned to ground truth class and Eqs. (44–45)) along with percentiles (over scenario realizations) as patches of varying saturation, with a 10% step: median is plotted darkest, the patch around it contains the runs between 40th and 60th percentile, the next one between 30th and 70th, etc. The area above and

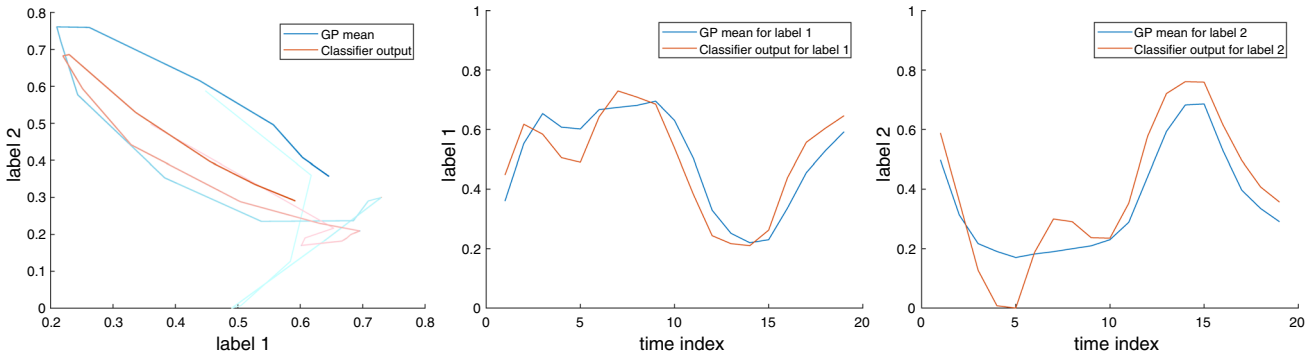


Fig. 2 Scenario with model uncertainty (no localization errors). Standard deviation for the noised observation was chosen in [0, 0.3]. Left: the ground truth class GP model mean and simulated (noised) classifier measurement over robot trajectory, plots of response for 1st label against response for 2nd label. More intense color corresponds to later time index. Center and right: first and second components over time indices, respectively

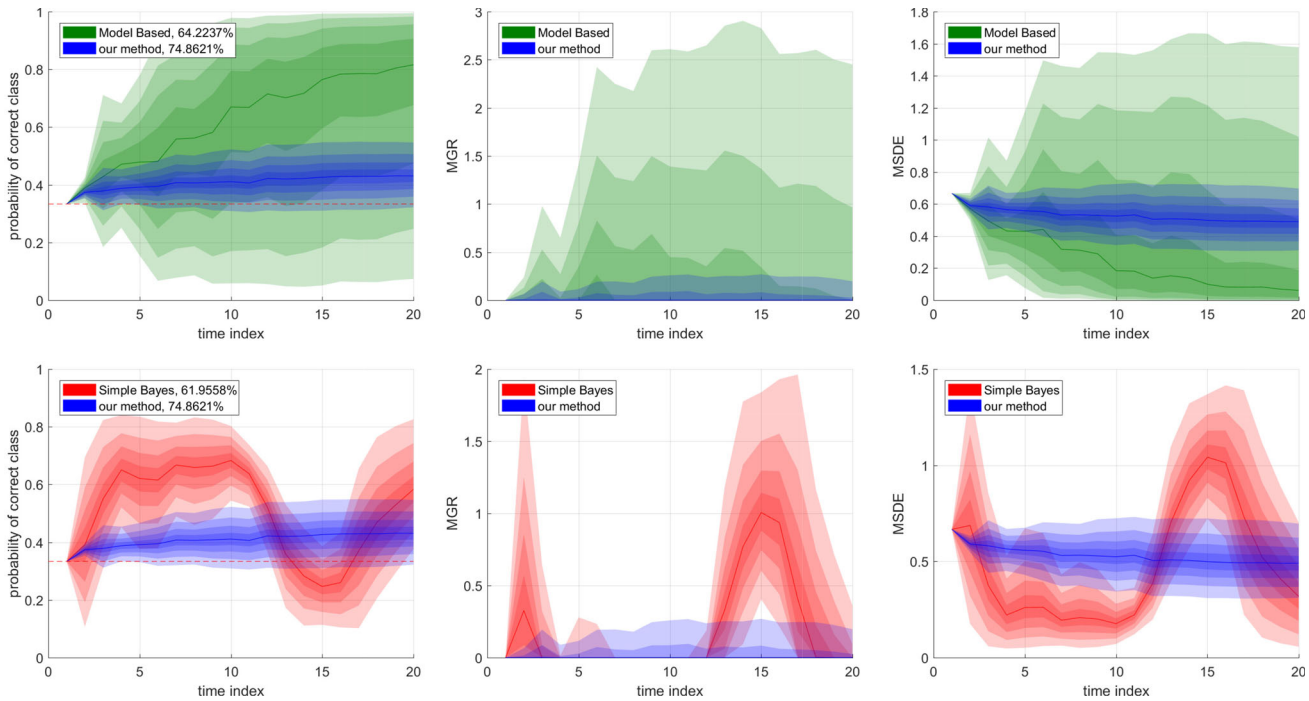


Fig. 3 Model Uncertainty synthetic scenario. Left column: probability of correct class, middle: MGR, right: MSDE. Legend in the leftmost column shows percentage of time steps where most likely class was the correct one. Color patches denote percentiles of the respective methods, one step in lightness denotes 10% percentile step and median is plotted darkest, so that the patch around the median comprises values between the 40th and the 60th percentile, the next darkest between the 30th and 70th and so on. Dashed line denotes the uninformative prior (probability of 1/3 for label) Top row: comparison of our method to Model Based, bottom row: to Simple Bayes. While probability of correct class in our method rises slowly, it fluctuates significantly less over realizations, and the correct class is chosen more often. In both plots, Simple Bayes method performs poorly where an "inverse" measurement (see Sect. 5.3.1) exists in the model, around time index 15

below the plots contains the top and bottom 10% of the runs respectively. Top row shows comparison of our method (blue) to `Model Based` (green), bottom - to `Simple Bayes` (in red).

An immediate observation in comparison to `Model Based` (first row) is that our percentiles are more concentrated, which means that method results are more stable. For example, in more than 20% of the runs (bottom lightest patch and below), probability of correct class (left column) for `Model Based` in time step 15 is less than 0.2 (compared to more than 0.33 for ours). Indeed, in more than 20% of the runs the MGR (middle column) for `Model Based` at iteration 15 is higher than 1, which means that a wrong (most likely) class was assigned probability more than twice

Algorithm 2 MATLAB simulation: procedure for simulating classifier outputs at step k

Input: $\mathcal{S}_{0:k-1}, \mathcal{X}_{0:k}^{(rel)}, \sigma_{max}^2, N_{samples}$
1: $s^{nominal} \sim \mathbb{P}(s \mid c, \mathcal{S}_{0:k-1}, \mathcal{X}_{0:k}^{(rel)})$ ▷ See Eq. (31)
2: $\sigma_u^2 \sim Uni(0, \sigma_{max}^2)$ ▷ Choose uncertainty level
3: $s^{noised} \sim N(s^{nominal}, \sigma_u^2 I)$ ▷ Uncertain classification
4: $samples \leftarrow \emptyset$
5: **for** $N_{samples}$ times **do** ▷ Simulating dropout
6: $\quad s \sim N(s^{noised}, \sigma_u^2 I)$
7: $\quad samples \leftarrow samples \cup \{s\}$
8: **end for**
9: **return** $s^{nominal}, s^{noised}, samples$

higher than the correct one, i.e. wrong class was chosen with high confidence. The MSDE plot displays similar behavior. In the bottom row, drop of accuracy of `Simple Bayes` around time step 15 is the result of an "inverse" measurement in the model, meaning that from a certain angle, classifier response suggests a different class (see for example in Fig. 1a). This illustrates well the difference from our method, which matches the entire sequence of measurements against a model, and thus can use also "inverse" measurements to classify correctly (on the downside, requiring a class model).

5.3.2 Localization uncertainty scenario

In methods making use of spatial class models, localization errors may cause classification aliasing when acquired measurements correspond to the model of a wrong class, because of the spatial shift in the query. To exemplify this, in this scenario, we introduced (a constant) bias in easting coordinate (the robot moves eastward in a straight line), causing aliasing between models (with no model uncertainty). Consider Fig. 4. The left plot as before shows GP mean of the correct class model (blue) and classifier output over robot track (red). It also shows the GP mean of the model of a wrong class (yellow). In the center plot, classifier outputs for label 2 (red) compared without localization bias against the corresponding GP component of the ground truth class model (blue) show a clear match. After introducing a bias of -8 units in easting (right plot) classifier responses (red) are matched against shifted spatial models, making the wrong class (yellow) a more likely match until around time step 16, after which the the blue line can be matched correctly in spite of the shift. The effects of this on performance are shown in Fig. 5. While our method, aware of the localization uncertainty (standard deviation) accumulates classification evidence gracefully, the `Model Based` method infers the wrong class with high confidence (as can be seen in the MGR plot, center) peaking at around time step 15, after which disambiguating measurements start to arrive. In the bottom row of the same figure, `Simple Bayes` method performs well (closely following the line from Fig. 4), since classifier measurements are stable and not ambiguous (the aliasing happens when trying to match against the different models).

5.4 Evaluation in synthetic 3D environments

To study our approach in a more realistic setting when using a DL classifier, we created a set of 3D simulated environments using UnrealEngine. In this setting, classifier measurements are obtained by feeding rendered images of an object to a neural network classifier (CaffeNet), in contrast to the basic MATLAB simulation, where they were generated from a manually specified GP model.

For simplicity, we limit ourselves to planar environments, containing a single object, corresponding to solved data association. Camera moves at a constant height of 160 cm, in object proximity - corresponding to object detection, in a way that the object occupies most of the frame—and facing it. We selected a set of objects for which 3D models were available, considering them as representative of the corresponding classes. For each object/class we fit a spatial (2D) GP model to classifier outputs for rendered viewpoints sampled over a grid. For simplicity, each view is directed towards the object.

Figure. 6 shows the GP models fit for each of the classes (chest, crate, desk), sampled over a 2D spatial grid centered on the object, as well as example images for corresponding viewpoints and raw classifications output by the classifier unit (as used for learning the GP models). The color of each pixel in the raster maps is calculated as the sum of class colors (blue for chest, orange for crate, green for desk), weighted by the classification vector predicted for the (object-centered) corresponding view.

We used classifier scores for the selected classes as features, although other choices are in principle possible, such as using scores for additional or fewer classes, or in fact any spatially varying feature, see discussion in Sect. 6.

We next show experiments using the above GP models for classification under model uncertainty, Sect. 5.4.1 and localization uncertainty, Sect. 5.4.2.

5.4.1 Model uncertainty experiments

In order to induce model uncertainty, we modified the crate 3D model by filling one of its sides with plain color. Note that while both the original and the modified crate were not part of classifier training set, we added color to induce an irregular appearance and elevated model uncertainty.

The spatial uncertainty map in Figure 7a shows the standard deviation of classifier vectors obtained from MC dropout for the corresponding views, summed over components (corresponding to the classes of interest). For all views camera is at a constant height, at each point oriented towards the object center. Data at the center of the map is

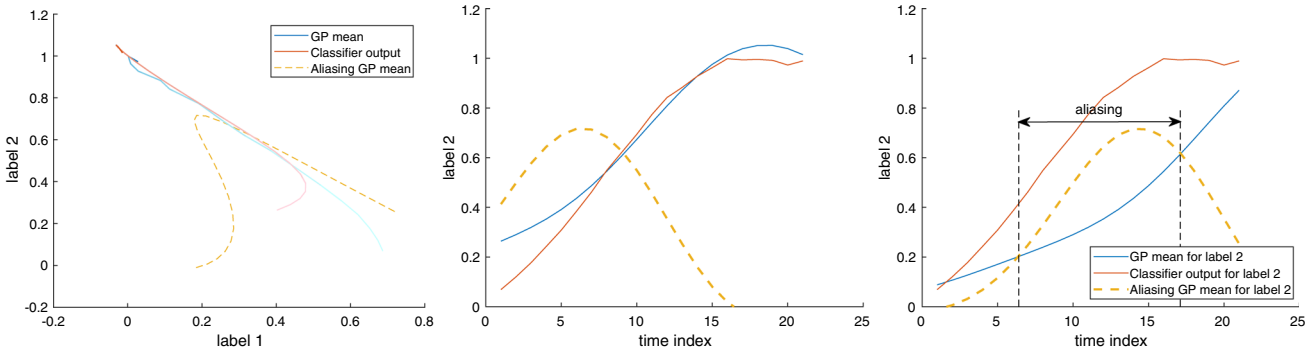


Fig. 4 Scenario with localization bias in the x axis. Time index corresponds to x coordinate (robot motion is a straight line, in the direction of the x axis). Left: as before, simulated classifier output generated from GP of ground truth class. Center: we concentrate on responses for class 2 (2nd component of classification vectors). Classifier output (red) matches GP of ground truth class (in blue) at true position. Right: bias in the x axis means that classifier output is effectively compared to a shifted model, better matching GP of a wrong class (yellow). This leads to classification errors unless accounting for localization uncertainty (Color figure online)

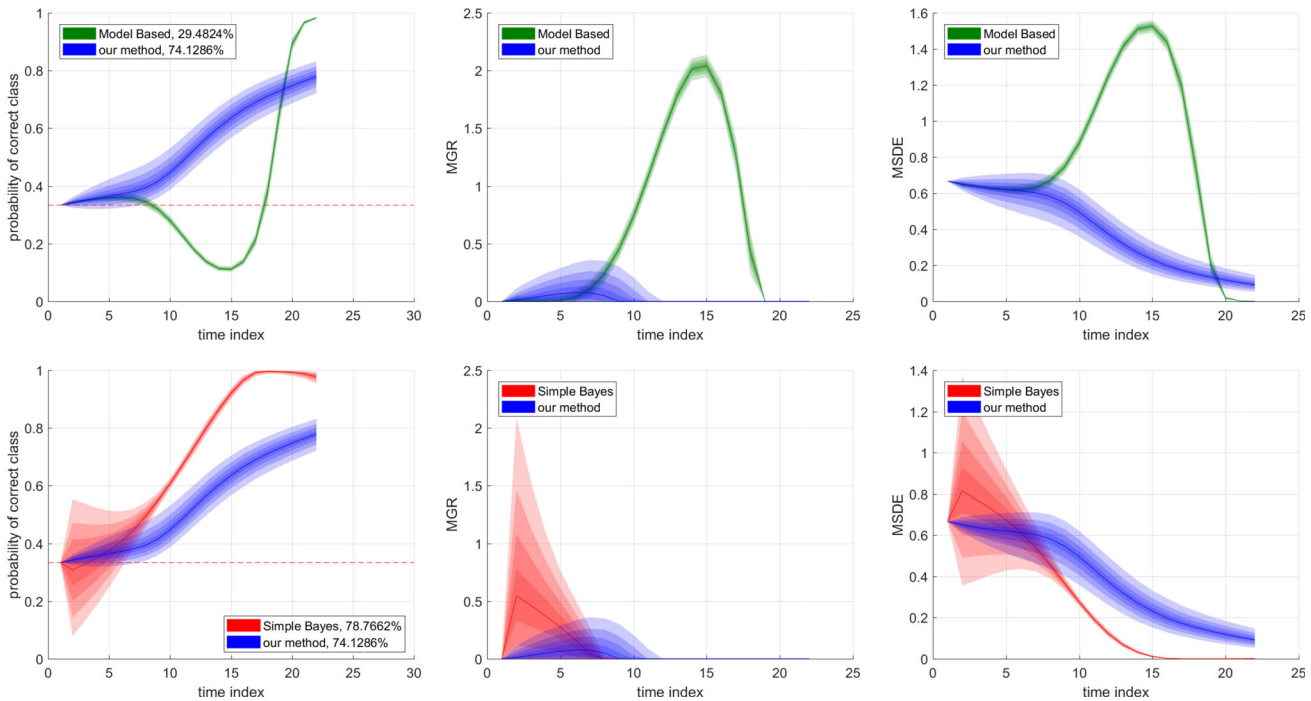


Fig. 5 Localization uncertainty synthetic scenario. Left column: probability of correct class, middle: MGR, right: MSDE. Localization bias of -8 units in the x axis causes severe aliasing in Model Based method resulting in a wrong class being inferred with high confidence. Our method is aware of localization uncertainty of standard deviation 16, and is able to recover. Simple Bayes method does not experience aliasing, as it uses the raw measurements directly, rather than matching them to a model

missing since views too close to the object become uninformative. As expected, the plot generally exhibits elevated uncertainty at points corresponding to views of the colored crate side (right side of the map). Uncertainty values also tend to be higher around the map edges (especially left center and corners), possibly related to larger portions of the crate's environment, which is different from training (natural) images, in the frame. Relatively low uncertainty values in the center right area are due to low values of components of the classification vector corresponding to classes of interest, which result in low variance.

We generated 300 random tracks in the area exhibiting elevated uncertainty values (right side of the map), rendering 10 viewpoints along each track. Two example tracks are shown in Figure 7a, example viewpoints are shown in Figure 7b. As before, viewpoints are centered on the object. For

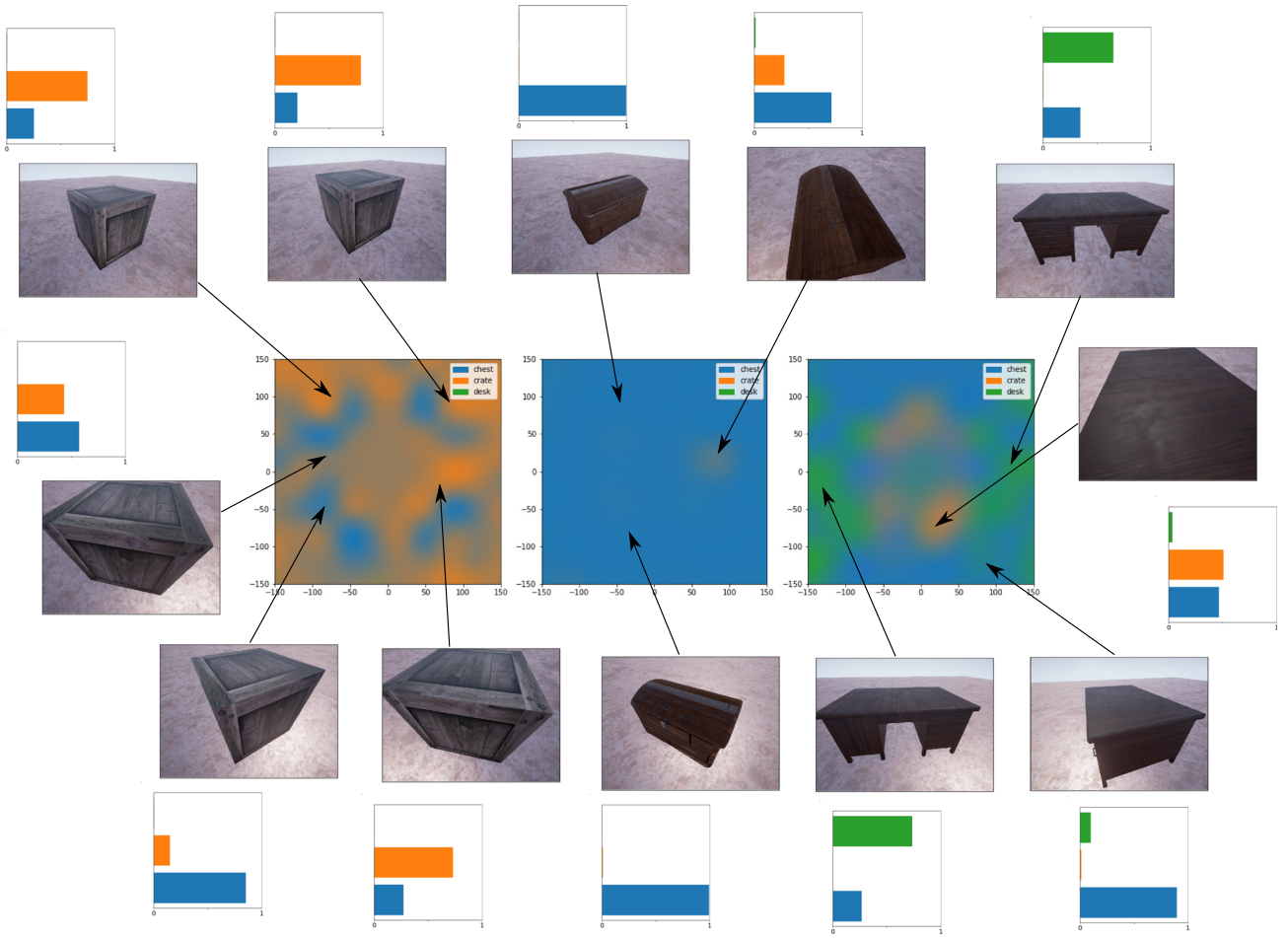
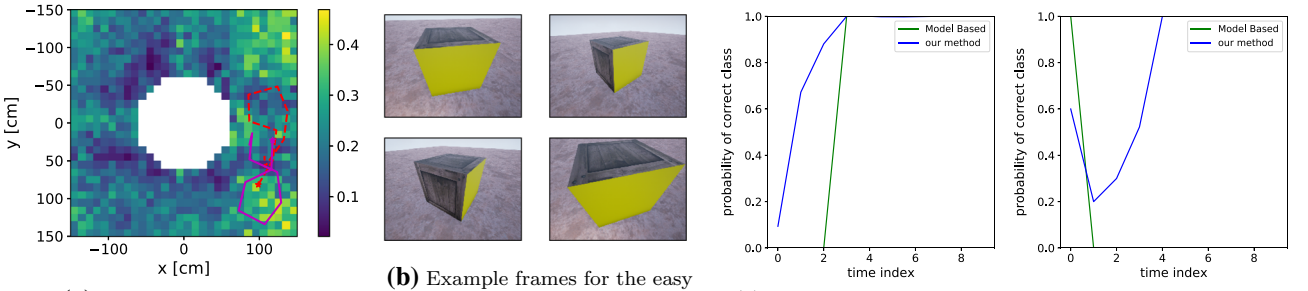


Fig. 6 Spatial GP models learned for the classes of interest and example views. Raster maps show spatial predicted classifications output by the black box classifier unit when observing an object of the corresponding class. Left raster shows predictions of GP model learned for a crate object, center - model for chest, right - model for desk. Each pixel corresponds to a classification vector predicted by the GP model. Class GP models are fit to "raw" classifier measurements for varying relative viewpoints. Thus, a strong orange component indicates that the element corresponding to the crate class in the classification vector output by the classifier is close to 1 for that viewpoint. For each model representative views are displayed along with classification vectors output by the classifier unit (bar plots). Note the difference between "raw" chest, crate and desk classifications output by the classifier unit, to chest, crate and desk class models which in general need not be related (i.e. to construct class models features other than explicit class predictions can be used). Learned models indicate that chosen crate object is for some viewpoints mis-classified as chest by the classifier unit, chosen chest object is generally classified correctly from all viewpoints, chosen desk object may be classified as each of the classes, depending on viewpoint, motivating the use of class models over raw classification scores

each frame, we estimated model uncertainty using 10 forward passes with MC dropout, although any other method can be used in the context of our Bayesian fusion scheme. In all experiments, class models, localization and classification measurements input to the compared methods are the same, but our method is aware of model uncertainty by using all MC-Dropout samples, while compared method is input only the samples average. Figures 7c and 7d compare classification of the two example tracks using `Model Based` and our method showing the evolution of score assigned to ground truth class (crate) over the time steps / track points. In the "hard" track (shown in red in Fig. 7a) the `ModelBased` method fails while ours is able to recover.

Figure 8 provides a statistical comparison of Bayesian classification using our method (in blue) against the `Model Based` method (in green). As before, color patches denote respective percentiles per-timestep over the random tracks. Plots show the evolution over time steps of probability assigned to ground truth class by each of the methods (left column) and the MGR (right column).

For both methods results vary over tracks. For the `ModelBased` method, median score for ground truth class was 1. However, for every time step in 20% of the tracks the ground truth class score is exactly 0 (threshold $= 10^{-10}$). For around 25% of the tracks, ground truth class score remains zero after measurements from the entire trajectory have been



(a) Model uncertainty map

(b) Example frames for the easy track

(c) Classification of easy track - magenta

(d) Classification of hard track - red

Fig. 7 Setting of the model uncertainty experiments. **a** Views of the colored side exhibit higher model uncertainty. Map shows standard deviation of classifier vectors obtained from MC dropout, summed over components. Two Example random tracks are plotted in red "hard" and magenta "easy". **b** Example frames along the easy track. **c** and **d** Probability of correct class over time steps along the track for easy and hard track respectively, using Model Based (green) and our method (blue) (Color figure online)

incorporated, compared to none for our method. Conversely, in our method, percentiles are rising steadily, corresponding to gradual accumulation of information, and are less spread out, in particular—not approaching score 0, which corresponds to a confident misclassification. This can be equally seen in the (log) MGR plot Fig. 8b, which for the `Model Based` method in up to 40% of the tracks reaches around 20 (meaning e^{20} times higher probability assigned to wrong class), and remains so at the end of 25% of the tracks (i.e. after incorporating data from the entire track), while for our method final MGR is 0 at 75% and 5.05 at 99%, implying reduced confident mis-classifications and thus relative resilience to model uncertainty.

Table 1 summarizes MGR (75th and 95th percentile) and correct most likely class rate, that is—ratio of tracks for which the ground truth class was the most likely. "Final" column presents statistics over tracks at the final time step, that is - for each track, after information from the entire track has been incorporated. "Overall" column presents statistics over all time steps from all tracks. Correct rates for both methods are similar, while MGR is considerably lower for our method, consistent with the plots.

This behavior also corresponds to results presented in Fig. 3 for the MATLAB simulation. While ground truth class scores for our method are generally lower than with `Model Based` method, reflecting the classification uncertainty, they are more stable under model uncertainty, and information is accumulated gracefully, as opposed to making overconfident mistakes.

5.4.2 Localization uncertainty experiments

To assess performance under a noisy relative pose estimate returned by SLAM / perception algorithm, we manually specified test tracks of varying classification difficulty (described below). Test tracks were then corrupted by perturbing each of the viewpoints along the track with an i.i.d. (both among tracks and among each track's viewpoints) Gaussian noise $N(0, \sigma)$, coordinate-wise. Each corrupted track thus obtained corresponds to one realization of a noisy robot trajectory and object localization estimate (while the ground truth remains the original test track before application of noise). Corrupted tracks were fed to the compared algorithms as the position estimate, our method additionally input σ as the estimate's uncertainty. In our experiments we collected results over realizations of corrupting noise, statistically exploring the effect of perception errors on classification of the chosen tracks.

Figure 9a shows the three test (ground truth) tracks over which statistics are presented in the below sections. The background shows the "aliasing map" in classification of a desk object under a (one particular) localization bias of +10 cm in the x axis, using the `Model Based` method (i.e., not uncertainty-aware). Each pixel is obtained by weighting the colors corresponding to the different classes (blue for chest, orange for crate, green for desk) with the classification vector obtained for the given viewpoint (as before, the object is at the coordinate origin). Thus, pixels with a strong orange component correspond to views for which the desk object is incorrectly classified as crate under the given localization bias, due to the raw classifier measurements for this view better matching the crate model.

Thus, the hard track (in red) passes (exclusively) through viewpoints which are incorrectly classified by the `Model Based` algorithm for this particular localization bias. The moderate track (black) passes through viewpoints that are classified correctly, but other points in their vicinity are not. The easy track (magenta) passes away from incorrectly classified viewpoints. Note that while we selected and described the ground truth tracks with respect to their difficulty under one particular value of localization bias, it turns out that they also exhibit the described behavior for sampled random val-

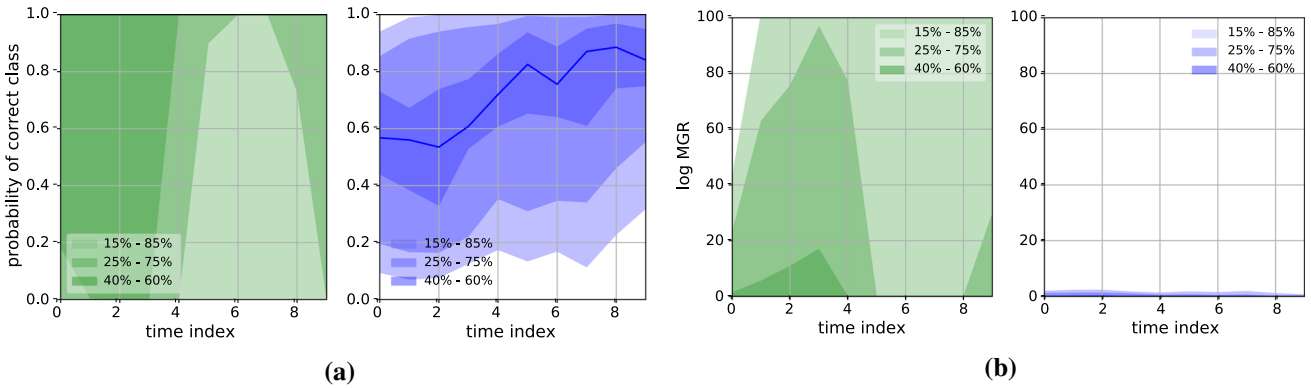


Fig. 8 Model uncertainty UE experiments. Blue: our method. Green: Model Based. Left (Fig. 8a): Probability assigned to ground truth class. Right (Fig. 8b): MGR, Eq. (45) (Color figure online)

Table 1 Model uncertainty experiment. Correct rate is ratio of classifications where the ground truth class was the most likely

	Correct rate-final	Correct rate-overall	MGR - final		MGR - overall	
			75%	95%	75%	95%
Model Based	0.74	**0.67**	29.98	629.2	44.67	519.30
Our method	**0.76**	0.65	**0**	**2.46**	**0.85**	**2.99**

Statistics are over tracks given measurements from the entire track ("final" columns, ratio of tracks) and given measurements up to every time step ("overall" columns, ratio of time steps from all tracks). Our method achieves roughly the same correct classification rates, but makes less confident mis-classifications, as reflected by the significantly smaller MGR
Bold denotes the best result for each column

ues of localization bias, generally different ones for each viewpoint along the track, as can be seen in statistical results described below.

The situation leading to classification aliasing is exemplified in Fig. 9b and c for classification of viewpoints along the hard track (red). To reduce clutter here we focus on the "chest" component of the raw classification measurement vector (dashed line), although in practice behavior is determined by all vector components at once, as shown in Fig. 10. In Fig. 9b, the "chest" component of the raw classification measurement vector (dashed) reasonably matches the desk class model predictions (plain line). While crate model appears to match the measurements better, in practice the other vector components provide disambiguation, as described in detail in Fig. 10. In Fig. 9c the same measurements (again, dashed) are compared with class models at a wrong location, due to the localization error, matching better the crate class model than the desk model, causing confident erroneous classification, for this particular bias, if not accounting for uncertainty. Figure 10 visualizes the entire classification vectors at each time step, providing insight into why there is disambiguation in the former (no localization error) but not the latter (with localization error) case.

Note that this effect closely corresponds to the MATLAB simulation, in particular what is presented in Fig. 4. Note however, that the experiments presented in the current section are different from the synthetic localization uncertainty experiments in Sect. 5.3.2, where tracks, localization bias and class models were engineered to illustrate contributions, and presented statistics were over realizations of the synthetic measurement models (GPs). In contrast, here classifier measurements both for training of class models and at test time are obtained from a Deep Neural Network classifier operating on rendered images, and statistics are presented over errors in perceived localization w.r.t. given ground-truth tracks.

Given Level of Uncertainty As described in Sects. 5.3.2, and 5.4.2, localization error causes class aliasing when the sequence of class measurements obtained over a track better fits a model other than the ground truth class when matched at the erroneous estimate. The effect of marginalization over pose uncertainty in Eq. (19) is averaging term (a) over the belief, term (b), i.e. matching obtained measurements against the models as above at different values of shift w.r.t. the localization estimate, according to the belief posterior distribution.

This averaging may mitigate class aliasing if the ground truth class is matched for at least a portion of the shift values under the belief. On the other hand, this averaging may reduce confidence of classification even when there is no aliasing due to localization errors, if aliasing happens for some portion of averaged shift values.

In Fig. 11 we explore statistics for a given level of uncertainty $\sigma = 10\ cm$ over random additive noise as described

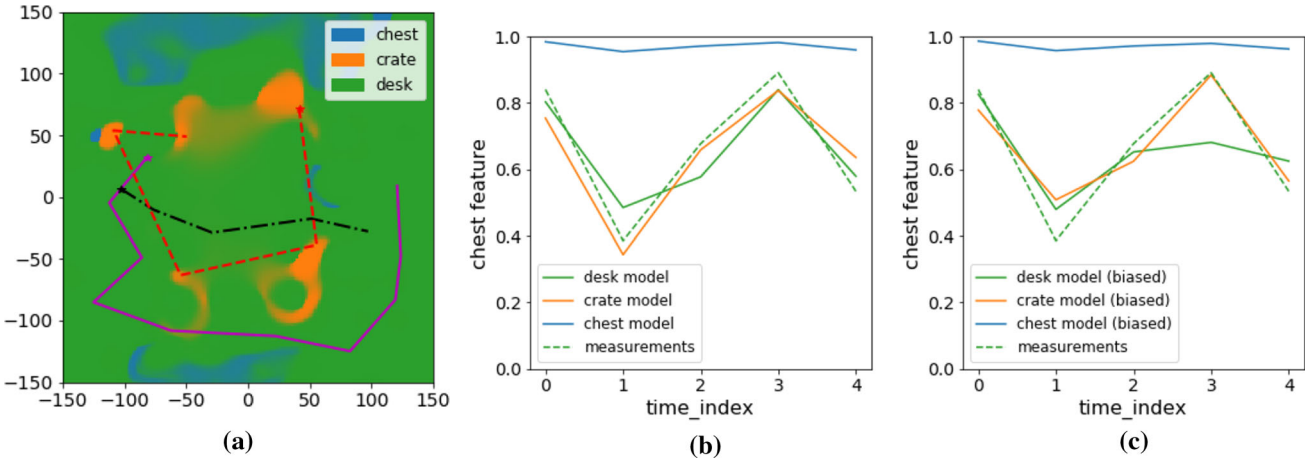


Fig. 9 **Left**: tracks of varying classification complexity, used for statistics presented in Fig. 11 against aliasing map for desk class, for bias of +10 cm in the x axis. The aliasing map is obtained by comparing per-pixel (every $\sim$ 5 cm) predictions from the desk GP model against shifted class models (see Figs. 6 and 4), corresponding to a localization bias, or a single realization in the localization uncertainty experiments. Magenta track (Fig. 11e and f, "easy") steers clear of aliasing areas making correct classification easy. Red track (Fig. 11a and b, "hard") passes through aliasing areas, causing classification errors in the Model Based method and high uncertainty in our method. Black track (Fig. 11c and d, "moderate") does not pass through aliasing areas for the particular bias pictured above, but does for other bias realizations, as can be seen in the classification statistics. Center and Right: class aliasing along the red track when viewing a "desk" object, equivalent of Fig. 4 for measurements and class models based on DL classifier output. Center plot shows the evolution of "chest" feature component in raw classification vectors (dashed line) against model predictions when localization estimate is correct (i.e. unlike the aliasing map shown in the left plot). Measurements curve appears to match both crate and desk class models (orange and green respectively), in practice inferred classification is mostly correct ("desk") as disambiguation is provided by other components (not shown in the plot). Figure 10 details the classification process. Right plot shows the same measurements curve (dashed) against models sampled with an error of +10 cm in the x axis (corresponding to the aliasing map in the left plot). Measurements match the crate model, leading to inference of wrong classification (see details in Fig. 10)

above (i.e. additive localization errors drawn from $N(0, \sigma)$ for each viewpoint) for the tracks from Fig. 9a, which turn out to exemplify both cases. The figure shows percentiles of probability assigned to ground truth class by the two methods (left two columns) and the MGR (right two columns) for each time step along the track for the hard (top row), moderate (middle row) and easy (bottom row) ground truth tracks, statistics taken over additive localization noise as described above.

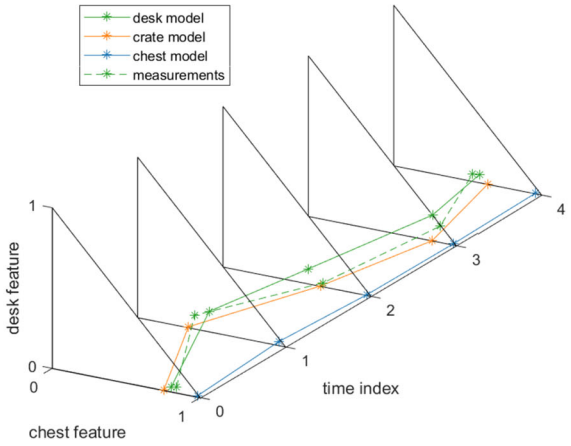
In the hard (Fig. 11a, b) and moderate (Fig. 11c and d) cases, our method mitigates aliasing, significantly reducing classification variability. The MGR plots indicate that mis-classifications, when they occur, are accompanied by uncertainty in the output classification, i.e. the correct class being assigned a non-negligible weight—note that the lightest patch indicates maximum values (i.e. 100th percentile), with lower percentiles much smaller. In contrast, confident mis-classifications occur in significant portions of the cases for the `Model Based` method, with the median MGR higher than 0 for the hard track. These results are equally supported by Table 2, which lists correct most likely class rates over all time steps ("Overall") and last time step ("Final") and 95th percentile of MGR for statistics with each of the test tracks. Our method reaches a similar, if slightly higher, correct rates as `Model Based`, with a much lower MGR.

In the easy case (Fig. 11e and f) our method more hesitantly reaches the same correct classification assigning lower confidence to the measurements due to uncertainty, with an identical rate of correct classifications overall.

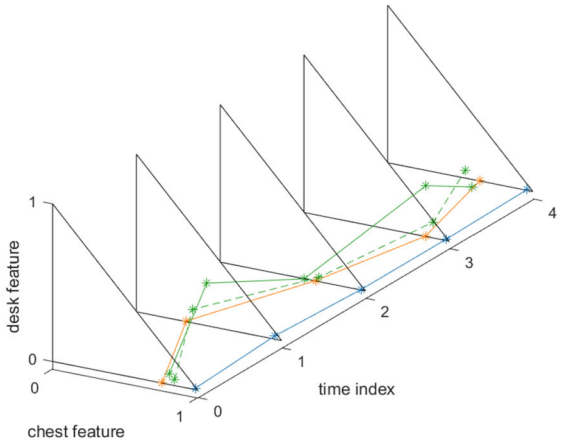
Sensitivity Experiments We explore the sensitivity of localization uncertainty aware classification to the level of localization uncertainty. To this end, we repeat the experiment from Sect. 5.4.2 for values of σ between 1 cm and 20 cm. Note that as the actual localization errors are drawn from a Gaussian distribution centered at 0, a higher value of σ does not imply that localization errors are necessarily higher, although higher errors do become more likely. As before, in each experiment our method is input σ in addition to the noisy localization estimate as measure of the localization uncertainty.

Figure 12 shows percentiles of probability assigned to ground truth class at the end of each track ("final") - left two columns, and at each time step using measurements up to that time step ("overall") - right two columns, both function of uncertainty level, for the hard track (top row), moderate track (middle row) and easy track (bottom row). Figure 13 shows corresponding percentiles of the MGR.

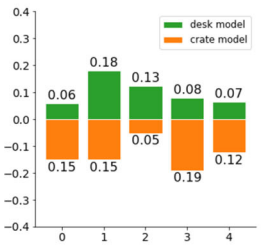
The plots generally exhibit behavior similar to the one observed in Sect. 5.4.2 for $\sigma = 10$ cm, this time for a range



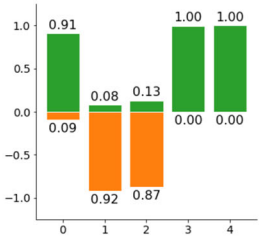
(a) Measurements and model predictions for correct localization



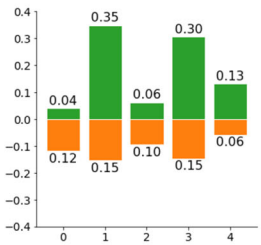
(b) Measurements and model predictions under localization error



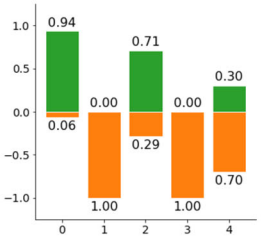
(c) Difference norm



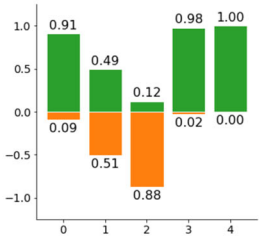
(d) Classification (single measurement)



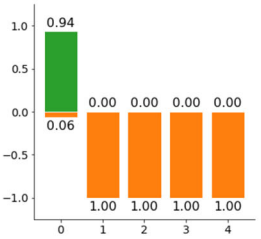
(e) Difference norm



(f) Classification (single measurement)



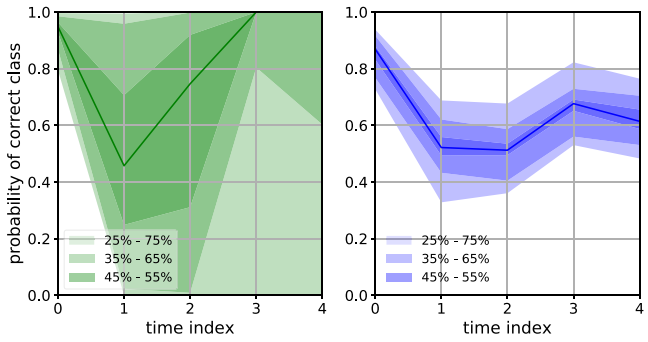
(g) Classification posterior (joint)



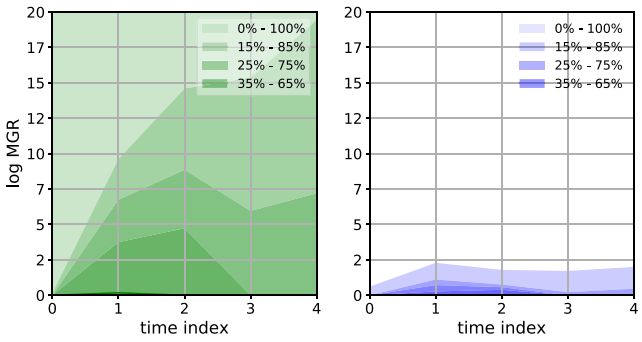
(h) Classification posterior (joint)

Fig. 10 Classification of a desk object over the red track from Fig. 9 under correct localization (left column) and with localization bias of +10 cm in the x axis (right column), corresponding to the scenario from Fig. 9. Top row: obtained measurements and class models samples over the track time steps. For every time index, raw classification vector (vertex on the dashed line) and model prediction mean for each candidate class are shown as points in the 2D simplex, corresponding to 3-coordinate classification vectors. Classification inference roughly corresponds to selecting the model curve which best fits the measurements. Figs. 10c and 10e show the Euclidean distance of measurement vector from prediction mean for desk and crate models, for each time index. The smaller the distance of measurement to model prediction, the more likely is the corresponding class. Fig. 10d and f show the corresponding class probability (i.e. normalized class likelihood) using a single measurement Eq. (5), whereas Fig. 10g and h show classification at each time index using all measurements obtained up to that time Eq. (7). While correct model (desk) does not perfectly match all obtained measurements (particularly at time indexes 1 and 2), eventually disambiguating measurements arrive and correct class is inferred in the left column. Introducing a localization error (right column) further brings measurements relatively closer to wrong model, causing the wrong class to be inferred Fig. 10h

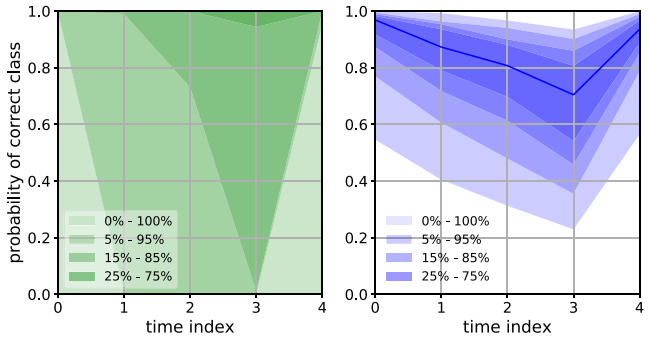
of σ values. As generally confidence of classification (probability assigned to ground truth class) is lower for our method, its percentiles are more concentrated and so results are more stable across the runs. Our method is also significantly less likely to assign a high probability to a wrong class, as demonstrated by the MGR plots Fig. 13.



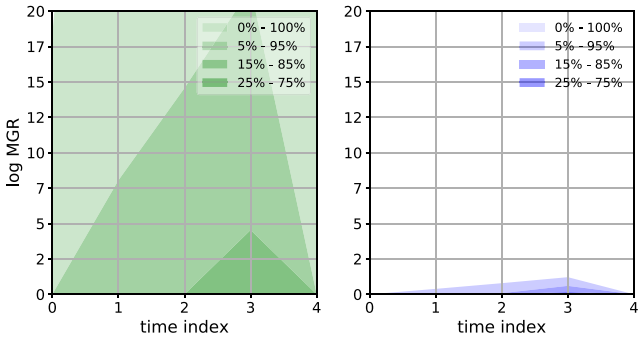
(a) Ground truth probability (hard case).



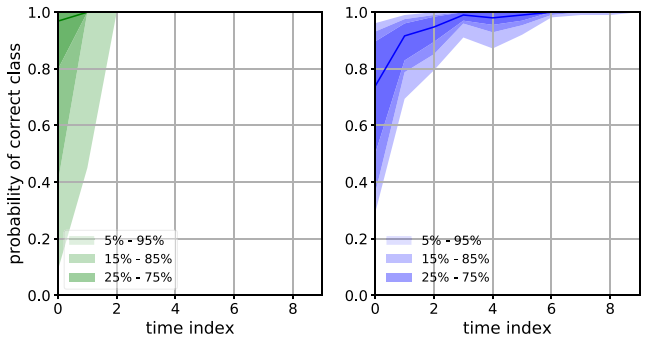
(b) Log MGR (hard case).



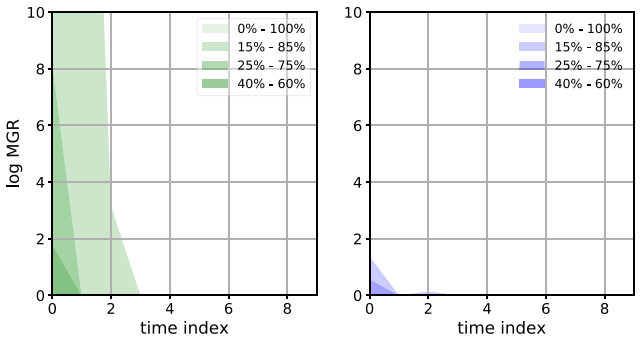
(c) Ground truth probability (moderate case).



(d) Log MGR (moderate case).



(e) Ground truth probability (easy case).



(f) Log MGR (easy case).

Fig. 11 Localization uncertainty UE experiments, for set level of uncertainty $\sigma = 10cm$. Localization estimate for both methods produced by corrupting test tracks with with an i.i.d. (both among tracks and among each track's viewpoints) Gaussian noise $N(0, \sigma)$, coordinate-wise. Our method is additionally input σ as the localization uncertainty measure. Blue: our method. Green: Model Based. Left column: Probability assigned to ground truth class. Right column: MGR, Eq. (45). Top row: hard track (Fig. 9), middle row: moderate track, bottom row: easy track

5.5 Evaluation with real-world imagery

We present results validating our contributions w.r.t. localization uncertainty in a real-world environment with the Active Vision Dataset (AVD) Ammirato et al. (2017), using data from the Berkeley Instance Recognition Dataset (BigBIRD) Singh et al. (2014) to learn GP models. Similarly to before, raw classifier measurements are provided by an AlexNet neural network classifier Krizhevsky et al. (2012) implementation in PyTorch Paszke et al. (2017).

The AVD comprises a set of indoor scenes, each one with images taken over a grid of spatial locations and camera orientations, allowing to simulate real-world trajectories. Each scene contains a set of BigBIRD object instances, with bounding boxes provided for each object in the set for each containing frame.

Table 2 Localization uncertainty scenarios—ratio of correct classifications, i.e. classifications where the ground truth class was the most likely given measurements from the entire track ("final" columns, ratio of tracks) and given measurements up to every time step ("overall" columns, ratio of time steps from all tracks)

	Easy				Moderate				Hard			
	Correct rate		MGR (95%)		Correct rate		MGR (95%)		Correct rate		MGR (95%)	
	final	overall	final	overall	final	overall	final	overall	final	overall	final	overall
Model based	1.0	0.97	0	0	0.958	0.90	0	8.8	0.66	0.64	48.6	29.0
Our method	1.0	**0.974**	0	0	**1.0**	**0.946**	0	**0.007**	**0.72**	**0.70**	**1.1**	**1.26**

As in the model-uncertainty experiments, uncertainty-awareness results in correct classification rates similar to the non-aware case, but significantly reducing rate of confident mis-classifications, see MGR in Fig. 11
Bold denotes the best result for each column

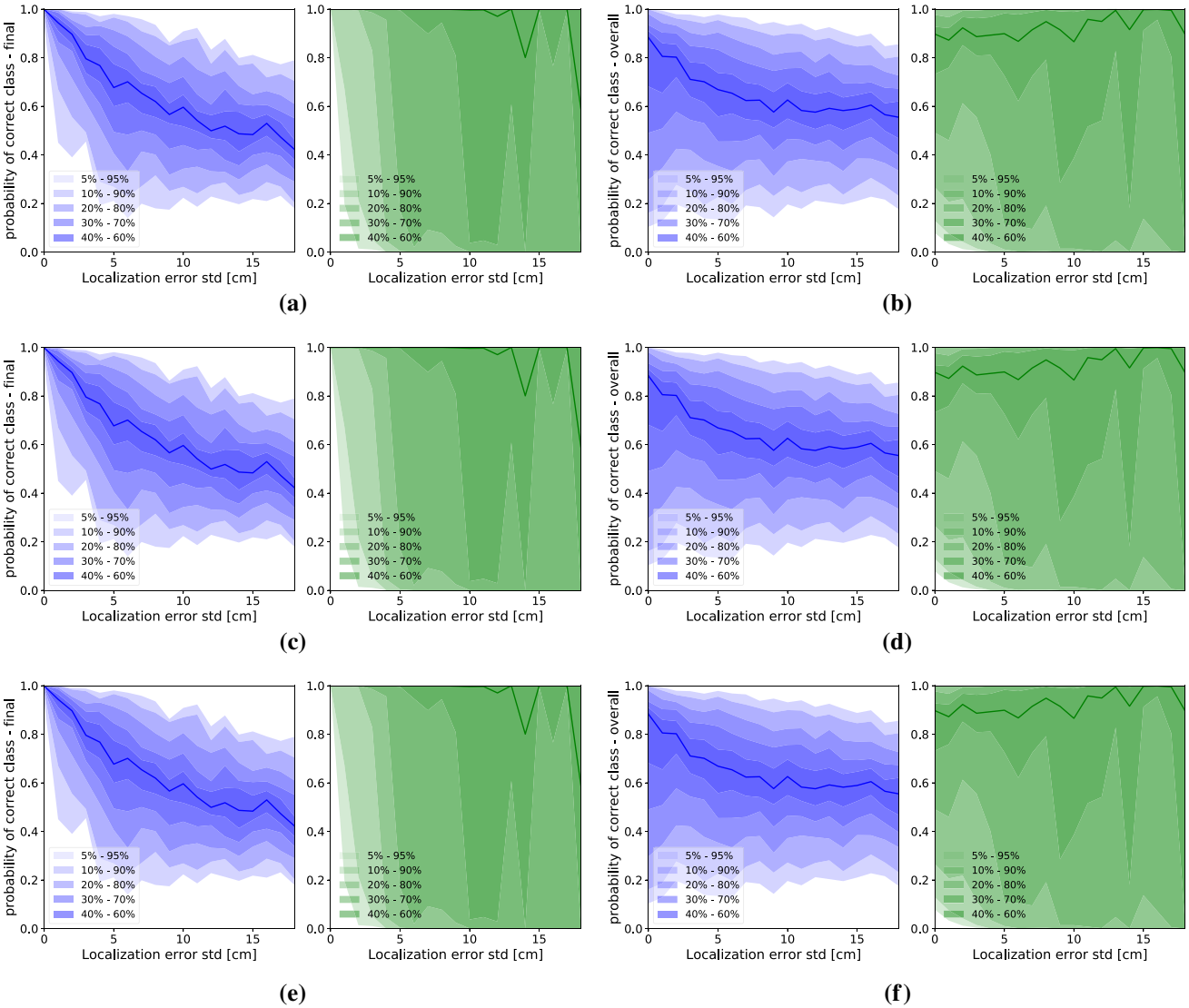


Fig. 12 Localization uncertainty sensitivity UE experiments, results for varying levels of localization error (σ). Probability assigned to ground truth class. Blue: our method. Green: Model Based. Left column: statistics over final time index. Right column: statistics over all time steps (see Table 2 and Eq. (46)). Top row: hard track, middle row: moderate track, bottom row: easy track

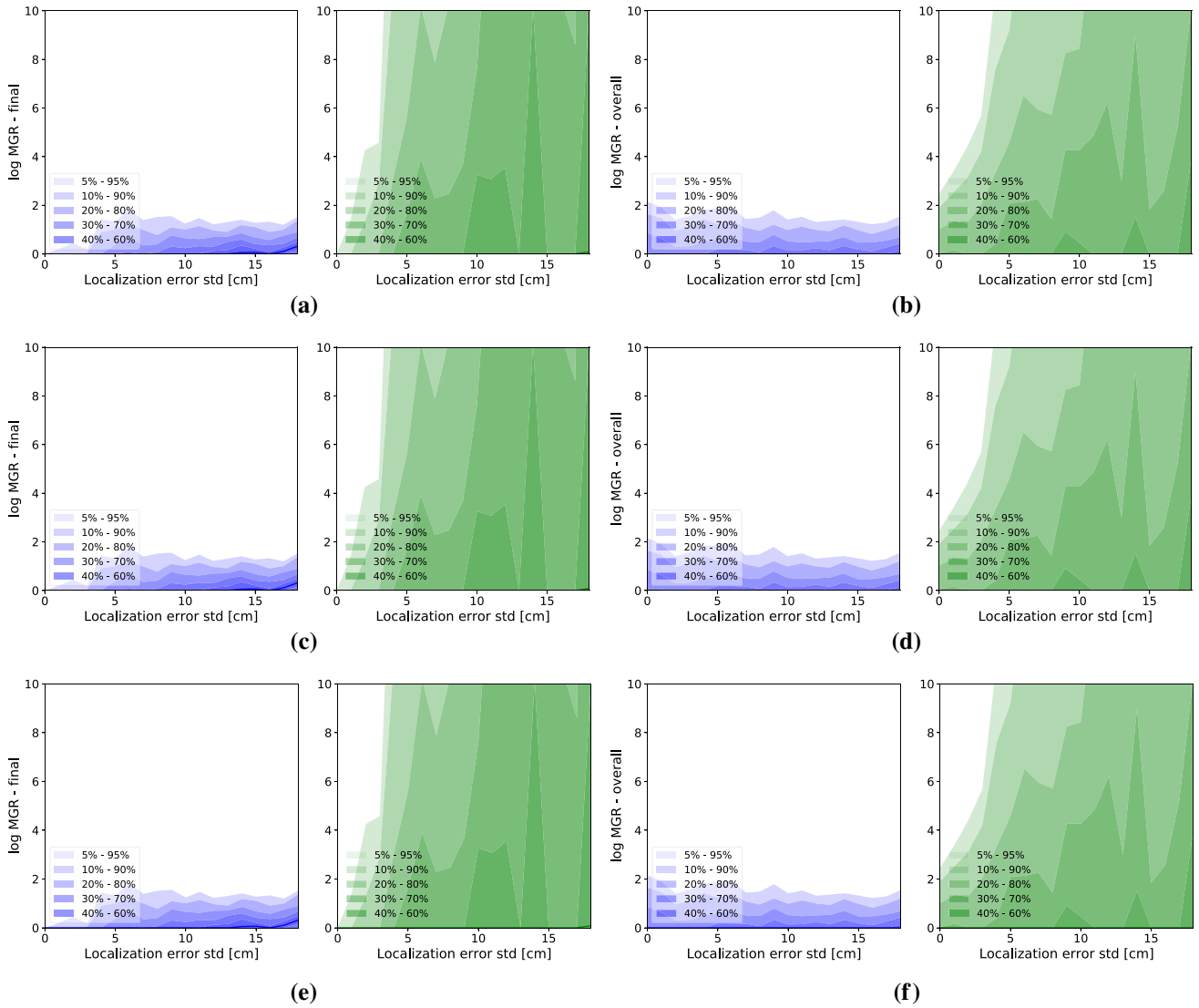


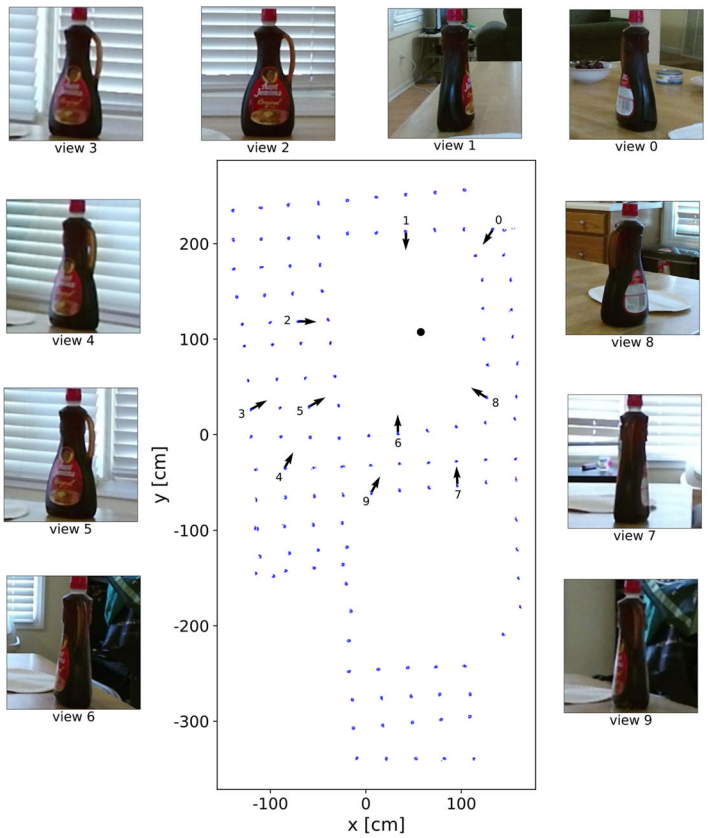
Fig. 13 Localization uncertainty sensitivity UE experiments, results for varying levels of localization error (σ). MGR, Eq. (45). Blue: our method. Green: Model Based. Left column: statistics over final time index. Right column: statistics over all time steps (see Table 2 and Eq. (46)). Top row: hard track, middle row: moderate track, bottom row: easy track (Color figure online)

For our initial evaluation, we focus on classification of an "aunt jemima original syrup" instance in scene `Home_001_1`. Figure 14 shows the scene layout. On the left (Fig. 14a) all viewpoints in the scene are denoted as blue dots. A subset of viewpoints chosen to demonstrate contributions is numbered with time step indexes as appear in later plots, arrows showing the ground truth camera orientations, pointed roughly towards the object, which is denoted by a large black dot. On the right, the frame at time 0 (top Fig. 14b) and time 4 (bottom Fig. 14c) are overlayed with object bounding boxes as provided in the dataset.

5.5.1 Learning spatial GP models with BigBIRD data

The BigBIRD dataset Singh et al. (2014) provides images of a set of "object instances" taken from varied angles. Images were taken using a set of static cameras with imaged objects placed on a rotating table Fig. 15. Ground truth poses for the cameras and the rotating table are provided, as well as a binary segmentation mask for each frame marking pixels belonging to the object.

We learn a spatial GP model for each object in the BigBIRD dataset, treating it as a separate class and using the provided segmentation masks to compute bounding boxes (Fig. 16 top row). These are then cropped and fed to an AlexNet to produce a set of classification/feature vectors,



(a) Layout of AVD scene Home_001_1 with all scene viewpoints shown in blue, and chosen views numbered. Around appear cropped object detections for the corresponding views. Black dot denotes (estimated) object location. Full frames for views 0 and 4 are given in Fig. 14b and Fig. 14c respectively.

(b) Frame at index 0 (from Fig. 14a) with bounding box corresponding to object of interest

(c) Frame at index 4 (from Fig. 14a) with bounding box corresponding to object of interest (view opposite to Fig. 14b

Fig. 14 AVD evaluation setting. Figure 14a shows the scene layout, viewpoints chosen for evaluation and corresponding object detections. Figure 14b and c show two example views of the scene

associated to corresponding camera poses relative to the rotating table, which can be directly computed from ground truth data. For simplicity, we limit ourselves to 2D models, implying classification of upright objects—which is enough for this evaluation. Accordingly, we only use features and corresponding relative poses computed for images from the bottom ring of relative camera poses in Fig. 15c. We fit GP models Eq. (7) to capture the spatial variation of chosen (elaborated below) classification vector components.

In Fig. 16 the second row from the top shows the GP model mean learned for each instance, as sampled at training set points, for a chosen subset of 4 classification vector components. As before, each component is assigned a color (as listed in the legend, along with component indexes), the color of a pixel in the raster is calculated by weighting the component colors with the corresponding values predicted by the model. Similar colors in raster plots correspond to similar raw classification vectors (components), thus indicating aliased (single) views and motivating both the accounting for (relative) localization uncertainty and the fusion of measurements from multiple viewpoints for disambiguation.

The third row from the top shows rasters of the same GP models, this time over the unit square, similar to Fig. 6. The bottom row shows rasters of GP models learned for a different set of components (again, elaborated below). BigBIRD object instances generally do not have an adequate corresponding ImageNet "ground truth" class, yet - as demonstrated by our evaluation - scores of an ImageNet-trained classifier can be directly re-used by our method as features to correctly classify an object fusing measurements from multiple views.

Choice of features The AlexNet classifier outputs scores for 1000 ImageNet Russakovsky et al. (2015) categories. While using the entire classification vectors to learn GP models is possible (especially under the independent components approximation Sect. 5), it is inefficient Sect. 4.4.1, and more so as the vast majority of components are negligible for most

objects. We thus choose to model 4 raw classification components with high (in particular, not negligible) scores across views for the object instances we used as classification candidates. The learned GP models are shown in the second and third rows of Fig. 16. Other choices are possible (e.g. bottom row of Fig. 16). The choice of good features is a research topic in its own right and is beyond the scope of this work.

GP parametrization/coordinates As BigBIRD data is only available on a ring (semi-sphere in 3D), images taken from front object views, we parametrize the GP models Eq. (7) with (uncertain) 2D location relative to the object, projecting the (relative) coordinates to the unit circle, that is, for the sake of this explanation denoting robot 2D relative pose to object o as $x_i^{(rel)} \doteq (t_{o\to i}, R_i)$ (i.e. $t^o_{o\to i}$ is robot 2D location w.r.t. the object), we approximate

$$\mathbb{P}(\mathcal{S}_{0:k} \mid c, \mathcal{X}^{(rel)}_{0:k}) \approx \mathbb{P}(\mathcal{S}_{0:k} \mid c, t^o_{o\to i}/\|t^o_{o\to i}\|). \tag{47}$$

Note that the above is an approximation, since in general classifier scores may be affected by distance from the object impacting e.g. the ability to distinguish detail due to finite resolution of input images. Additionally, an object viewed from farther away is more likely to be partially occluded and affected by limitations of the object detector, all of which could in principle be incorporated into the class model (but are out of scope of the present work). While distance from object could be simulated by purportedly blurring and subsampling input images, we did not go as far in the present initial evaluation.

Finally, it makes sense to consider parametrizing the class models with polar coordinates, which could allow to better capture different co-variation levels in the radial and tangential directions (partly visible in Fig. 6), to reduce calculations (e.g. in Eqs. (15), (16)) by reducing the dimensionality of GP inputs, and possibly to improve precision with a given amount of samples. On the downside, polar parametrization would entail domain considerations, such as ensuring continuity at boundaries. As using GPs for spatial modeling of classifier / detector scores (e.g. Teacy et al. (2015), Velez et al. (2012)) is not a contribution of this research (rather, the extension of the approach to handle localization and model uncertainty) we did not explore the various parametrization options in depth.

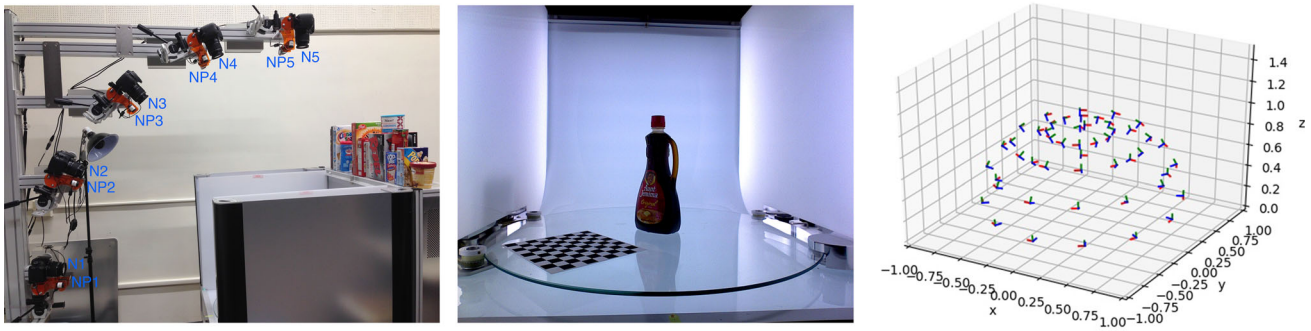
5.5.2 Evaluation in classification under localization uncertainty

In Fig. 17a we present the evolution of classification over the set of views treated as a track, and the object instance shown in Fig. 14 in a setting similar to Sect. 5.3.2, i.e. we introduce a bias corresponding to erroneous localization estimate to each pose along the track, producing an erroneous estimate of pose relative to the object. For classification candidates, we randomly chose a subset of 3 object instances (in addition to the ground truth class "aunt jemima original syrup"), shown in Fig. 16. The corresponding models display relatively little aliasing with the ground truth model (mostly in the second quadrant, with "crest complete minty fresh" and "red bull").

Both `Model Based` method and ours are fed with classification vectors from the true poses along with the erroneous pose estimate, our method in addition being input a standard deviation of $30cm$ for the 2D localization estimate (on each axis). The `Model Based` method is first affected before recovering, while our method, accounting for the pose uncertainty, is able to correctly accumulate information. To statistically examine this behavior, we randomize the localization estimate in the vicinity of the above biased estimate, adding to it a vector drawn from a zero centered Gaussian with standard deviation of $30cm$ along each axis. As before, our method receives a constant standard deviation of $30cm$ as the estimate uncertainty. The center and right columns of Fig. 17 show percentiles of the correct class/instance ("aunt jemima original syrup") probability output by each method (Fig. 17b) as well as the log MGR Eq. (45) (Fig. 17c) over the track / time indexes. Similarly to before, while in most cases both methods eventually deduce the correct class, the `Model Based` method experiences hard failures in around 10% of the runs outputting high probability for an incorrect class (around 10 times the score of the correct class) at the last step, while our method is able to classify correctly.

The reason for this behavior can be seen by considering variations in classification as function of localization bias, i.e.—given a raw classifier response—what is the GP model / class likelihood for various deviations of the localization estimate from ground truth. Computing this likelihood is equivalent to classification using the `Model Based` method (i.e., not considering localization uncertainty). Figure 18 shows the classification obtained for a range of bias in 2D localization at different steps along the track from Fig. 14a. For each plot, the axes denote the amount of localization error in centimeters, the value at (0, 0) corresponding to classification obtained with ground truth localization. The color at each pixel is obtained as the sum of colors corresponding to the object classes (shown in the legend), weighted by their likelihood given the localization error, e.g. pixel value at $(-30, 20)$ corresponds to classification obtained (by `Model Based`) given that localization estimate is off by $-30cm$ in the x axis, and $+20cm$ in the z axis, the correct class, "`aunt_jemima_original_syrup`", corresponding to blue color. For example, in view 0 the classification is correct for the above bias, whereas in views 3 and possibly also 6 classification is incorrect, which can be seen as the color departs from blue.

In the plot, colors of pixels indicate specific values of localization error causing the `Model Based` method to



(a) (b) (c)

Fig. 15 BigBIRD dataset setup (Fig. 15a, b taken from the dataset). Left: data collection setup, cameras facing the rotating table. Center: an example dataset image of the target instance on the rotating table. Right: Subset of camera poses relative to the rotating table for the different rotating table states. In all, images for 120 poses of the rotating table are provided, for each of the 5 cameras. We only use images from NP1 (i.e. the bottom row of relative poses in Fig. 15c) to learn 2D GP models, accordingly limiting ourselves (for simplicity) to classification of objects that are upright (w.r.t. BigBIRD coordinates)

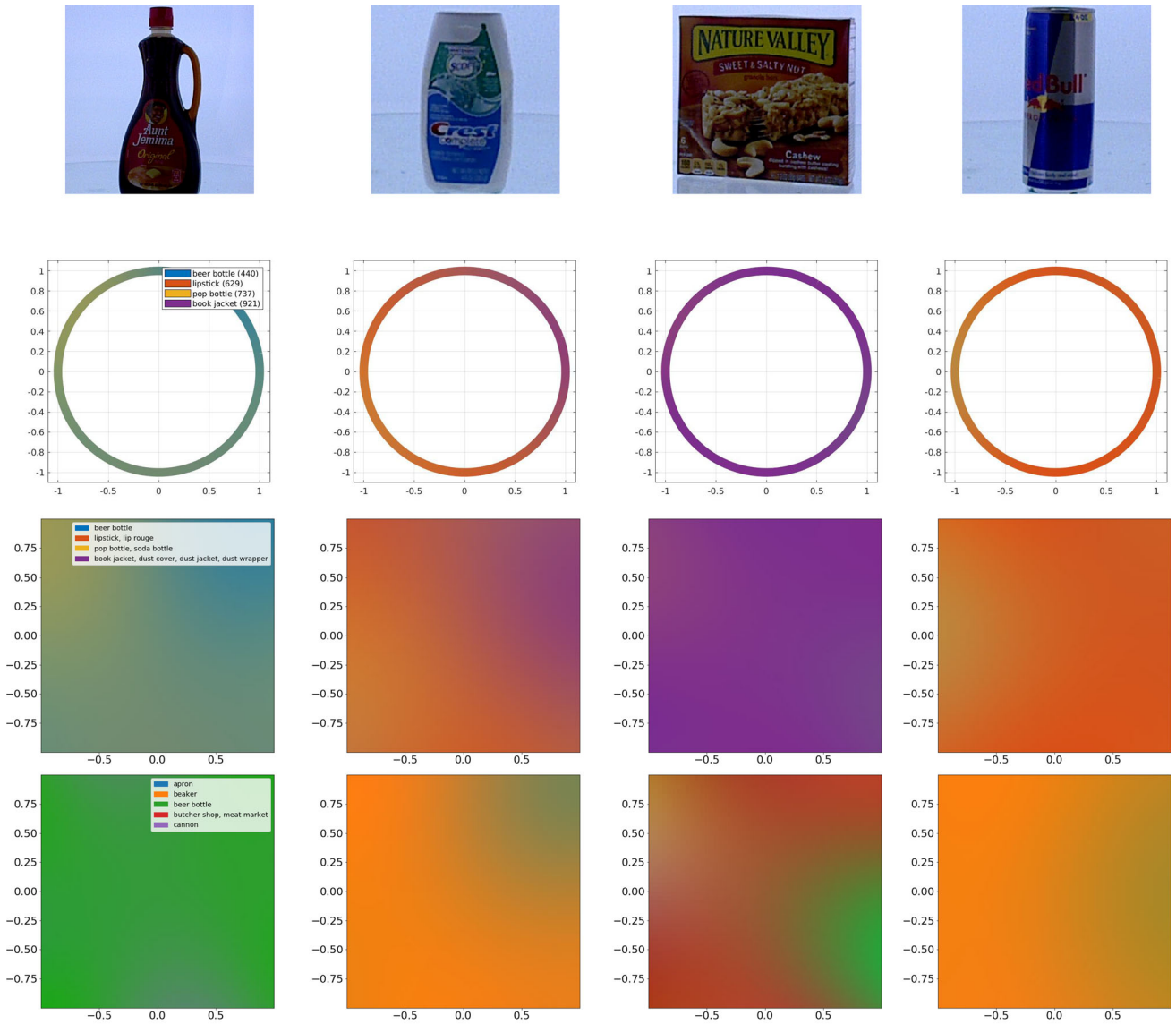
fail - the real-world equivalent of Fig. 4. Assuming model uncertainty negligible, the output of our method is obtained according to Eq. (42), roughly by averaging the output of `Model Based` over localization samples, drawn from a Gaussian centered at the erroneous estimate. Localization-uncertainty-aware classification score (i.e. using our method) thus depends on the proportion of samples of the correct color (the "average color") in the vicinity of each pixel. The wider spread of percentiles in Fig. 17 of the `Model Based` method is likewise explained by that its classification decisions are based on the value at a single localization sample, whereby bad estimates directly lead to erroneous classification ouputs. Averaging over a belief rather than using a single estimate allows us to mitigate the localization uncertainty-induced aliasing where enough localization samples produce the correct classification, and in any case - produces smoother classification decisions. On the downside, this sampling / averaging may reduce the confidence in the correct class where some of the samples produce incorrect classifications, as in view 1 in Fig. 17a.

6 Conclusions

We described a method for robust visual classification of an object of interest observed from multiple views under model and localization uncertainty, using a viewpoint-dependent semantic measurement model to capture viewpoint variability and spatial correlations in classifier scores and a Bayesian classifier providing a measure of model uncertainty to account for uncertainty due to out-of-distribution measurements in inference. We evaluated our method first in synthetic simulation, then in a 3D environment where rendered images were fed into a Neural Network classifier and finally on real-world data, comparing it to baseline methods in scenarios affected to varying levels by model uncertainty and localization uncertainty, in particular investigating in detail the behavior under varying levels of localization uncertainty w.r.t. the baseline method.

The results display a general improvement in resilience to localization error and to classifier noise due to out-of-distribution inputs w.r.t. not taking into account the uncertainty in the above. While correct classification rates are generally similar to the baseline method, our method accounts for the different sources of uncertainty, significantly reducing confident mis-classifications. Ability to operate under localization uncertainty allows deployment of our method in autonomous systems that need accurate semantic perception.

There are however fundamental limitations that must be noted and possibly addressed in further research. First, capturing correlations between multiple views requires marginalization over a track, which in our case is done by sampling. While sampling can be limited to poses from which the target object is observed, faithfully representing a distribution over multiple viewpoints using a limited number of samples is generally challenging, especially so as all marginal distributions of poses of interest change with each new view, thus a simple reuse of samples is not possible. A possible approach is to re-use most samples from previous time steps, weighting them correspondingly to the new localization posterior (importance sampling). In practical applications, the limited amount of samples may be less of a challenge, as heuristics can be used to approximate the joint distribution over multiple views, and massively parallel processing (e.g. GPUs) can be used to efficiently perform computations with sufficiently many samples. For further efficiency, viewpoint-dependent class models can be com-



(a) "aunt jemima original syrup"

(b) "crest complete minty fresh"

(c) "nature valley sweet and salty nut cashew"

(d) "red bull"

Fig. 16 Object instances and corresponding GP models. Top row: BigBIRD instances, cropped using provided bounding boxes. Second from top: GP models at training points. Third row: GP models raster. Bottom row: GP models for a different set of features (chosen as ones with highest significance with xgboost Chen and Guestrin (2016)). Instance names are listed in captions as they appear in the dataset

bined with non-viewpoint-dependent ones, to only describe salient objects for which estimating orientation is of interest. Another limitation, which is a general issue to class models, is that representative classifier responses need to be obtained for each class to be recognized, capturing all variability in appearance from all viewpoints, which moreover for a viewpoint-dependent model need to be known at training time as ground truth. As coming up with objects representative of an entire class is challenging, class models are perhaps better suited to recognizing particular objects or sets of objects which are known and can be measured in advance. Finally, the current method is an inference method, and as such it depends on externally collected measurements, possibly insufficient or in a sub-optimal way. Possible future work can involve developing a scheme to plan information collection in this setting, as well as rigorously address limitations mentioned above.

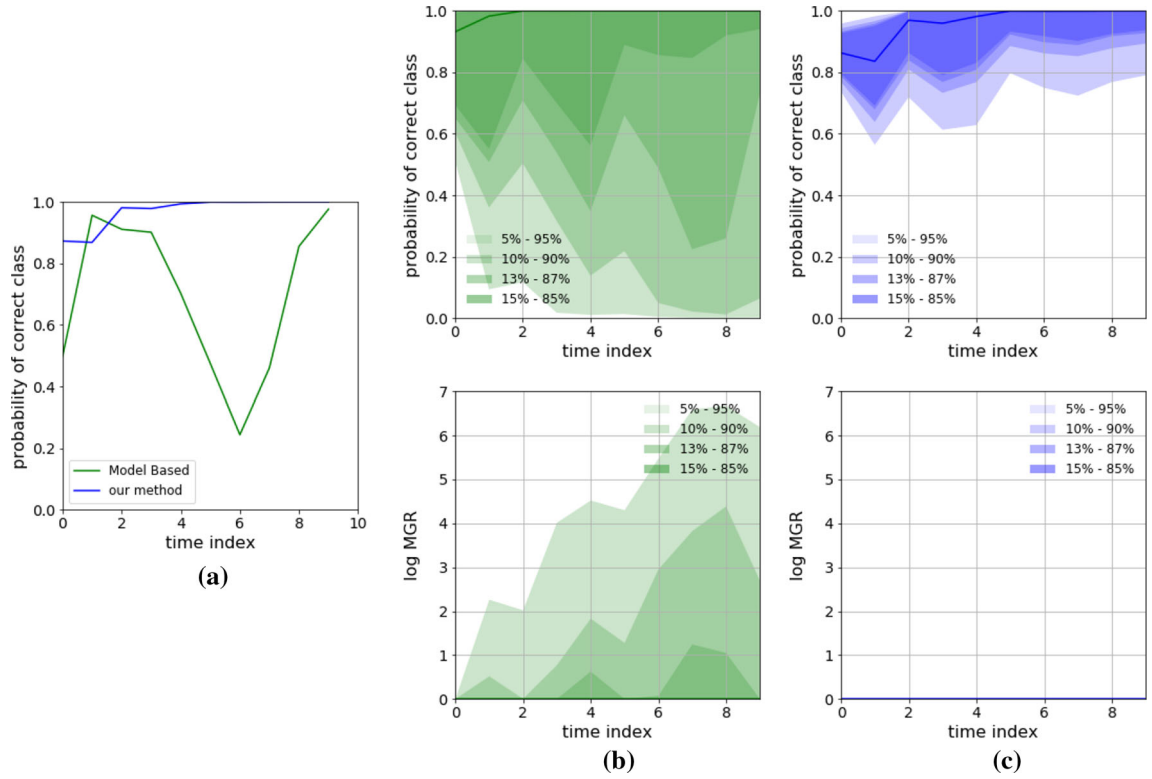


Fig. 17 Left: classification results (probability of ground truth class "aunt jemima original syrup") over the track from Fig. 14a with erroneous localization estimate for the `Model Based` and our method. Center and right: statistical results when randomizing the error in localization estimate in the vicinity of the error from before. As before, color patch denotes the respective percentile range. Top row: probability of correct class, bottom row: log MGR Eq. (45). As the `Model Based` method classifies based on a single localization sample, its results vary, while our method is more resilient to the noisy estimate, consistent with the behavior observed in simulation

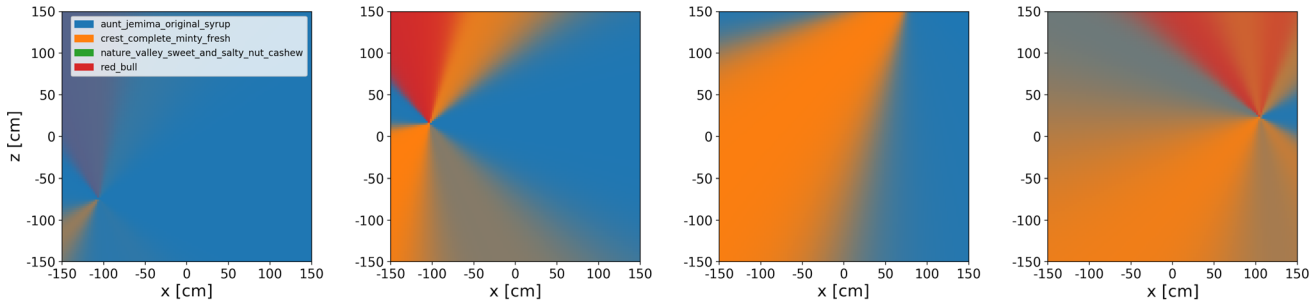


Fig. 18 Variations in classification function of localization bias for single views of the track from Fig. 14a. Each plot is obtained for a given semantic observation obtained from ground truth pose / viewpoint. The axes denote the amount of bias in centimeters, the color at each pixel is obtained as a (weighted) sum of colors corresponding to object classes (shown in the legend), weighted by their likelihood for the given localization error, i.e. the pixel at (0, 0) shows classification with localization estimate equal to ground truth position (Note the difference from the raster plots of Fig. 16, where colors correspond to raw classification vector components)

References

Ammirato, P., Poirson, P., Park, E., Kosecka, J., & Berg, A. C. (2017). A dataset for developing and benchmarking active vision. In *IEEE International Conference on Robotics and Automation (ICRA)*

Atanasov, N., Sankaran, B., Ny, J., Pappas, G. J., & Daniilidis, K. (2014). Nonmyopic view planning for active object classification and pose estimation. *IEEE Transactions on Robotics*, *30*, 1078–1090.

Becerra, I., Valentín-Coronado, L. M., Murrieta-Cid, R., & Latombe, J. C. (2016). Reliable confirmation of an object identity by a mobile robot: A mixed appearance/localization-driven motion approach. *International Journal of Robotics Research*, *35*(10), 1207–1233.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. *Machine Learning*, *79*(1–2), 151–175.

Bowman, S., Atanasov, N., Daniilidis, K., & Pappas, G. (2017). Probabilistic data association for semantic slam. In *IEEE International Conference on Robotics and Automation (ICRA), IEEE* (pp. 1722–1729).

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., et al. (2016). Simultaneous localization and mapping: Present, future, and the robust-perception age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Choudhary, S., Carlone, L., Nieto, C., Rogers, J., Christensen, H. I., & Dellaert, F. (2017). Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models. *International Journal of Robotics Research*, *36*(12), 1286–1311.

Farhi, E. I., & Indelman, V. (2019). ix-bsp: Belief space planning through incremental expectation. In *IEEE International Conference on Robotics and Automation (ICRA)*

Feldman, Y., & Indelman, V. (2018a). Bayesian viewpoint-dependent robust classification under model and localization uncertainty. In *IEEE International Conference on Robotics and Automation (ICRA)*

Feldman, Y., & Indelman, V. (2018b). Towards robust autonomous semantic perception. In *Workshop on representing a complex world: perception, inference, and learning for joint semantic, geometric, and physical understanding, in conjunction with ieee international conference on robotics and automation (ICRA)*

Gal, Y. (2017). *Uncertainty in deep learning*. Ph.D. thesis, University of Cambridge

Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning (ICML)*

Gal, Y., Islam, R., & Ghahramani, Z. (2017). Deep bayesian active learning with image data. In *International Conference on machine learning (ICML)*, JMLR. org (pp. 1183–1192).

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. arXiv:1408.5093

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J., & Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. *International Journal of Robotics Research*, *31*, 217–236.

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems (NIPS)* (pp. 5580–5590).

Kendall, A., Badrinarayanan, V., & Cipolla, R. (2015a). Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. arXiv:1511.02680.

Kendall, A., Grimes, M., & Cipolla, R. (2015b). Posenet: Convolutional networks for real-time 6-dof camera relocalization. In *International Conference on Computer Vision (ICCV)*.

Kopitkov, D., & Indelman, V. (2018). Robot localization through information recovered from cnn classificators. In *IEEE/RSJ International conference on intelligent robots and systems (IROS), IEEE*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems (NIPS)* (pp. 6402–6413).

Lianos, K. N., Schonberger, J. L., Pollefeys, M., & Sattler, T. (2018). Vso: Visual semantic odometry. In *European Conference on Computer Vision (ECCV)* (pp. 234–250).

Lütjens, B., Everett, M., & How, J. P. (2018). Safe reinforcement learning with model uncertainty estimates. arXiv:1810.08700

Malinin, A., & Gales, M. (2018). Predictive uncertainty estimation via prior networks. In *Advances in neural information processing systems (NIPS)* (pp. 7047–7058).

Malinin, A., Ragni, A., Knill, K., & Gales, M. (2017). Incorporating uncertainty into deep learning for spoken language assessment. In *Proceedings of the 55th annual meeting of the association for computational linguistics (Volume 2: Short Papers)*, vol. 2 (pp. 45–50).

McAllister, R., Gal, Y., Kendall, A., Van Der Wilk, M., Shah, A., Cipolla, R., & Weller, A. V. (2017). Concrete problems for autonomous vehicle safety: advantages of bayesian deep learning. In *International Joint Conference on AI (IJCAI)*.

Miller, D., Dayoub, F., Milford, M., & Sünderhauf, N. (2018a). Evaluating merging strategies for sampling-based uncertainty techniques in object detection. arXiv:1809.06006

Miller, D., Nicholson, L., Dayoub, F., & Sünderhauf, N. (2018b). Dropout sampling for robust object detection in open-set conditions. In *IEEE International conference on robotics and automation (ICRA), IEEE* (pp. 1–7).

Mu, B., Liu, S. Y., Paull, L., Leonard, J., & How, J. (2016). Slam with objects using a nonparametric pose graph. In *IEEE/RSJ International conference on intelligent robots and systems (IROS)*.

Myshkov, P., & Julier, S. (2016). Posterior distribution analysis for bayesian inference in neural networks. *NIPS: In workshop on Bayesian deep learning*.

Omidshafiei, S., Lopez, B. T., How, J. P., & Vian, J. (2016). Hierarchical bayesian noise inference for robust real-time probabilistic object classification. arXiv:1605.01042

Osband, I., Blundell, C., Pritzel, A., & Van Roy, B. (2016). Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems (NIPS)* (pp. 4026–4034).

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). Automatic differentiation in pytorch. *Advances in neural information processing systems (NIPS)*

Patten, T., Zillich, M., Fitch, R., Vincze, M., & Sukkarieh, S. (2016). Viewpoint evaluation for online 3-d active object classification. *IEEE Robotics and Automation Letters (RA-L)*, *1*(1), 73–81.

Patten, T., Martens, W., & Fitch, R. (2018). Monte carlo planning for active object classification. *Autonomous Robots*, *42*(2), 391–421.

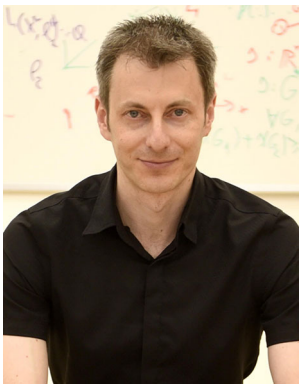
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Pillai, S., & Leonard, J. (2015). Monocular slam supported object recognition. In *Robotics: Science and Systems (RSS)*.

Qiu, W., Zhong, F., Zhang, Y., Qiao, S., Xiao, Z., Kim, T. S., & Wang, Y. (2017). Unrealcv: Virtual worlds for computer vision. In *Pro-*

ceedings of the 2017 ACM on multimedia conference, ACM (pp. 1221–1224).
Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2009). *Dataset shift in machine learning*. Cambridge: The MIT press.
Rasmussen, C., & Williams, C. (2006). *Gaussian processes for machine learning*. Cambridge: The MIT press.
Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252.
Salas-Moreno, R. F., Newcombe, R. A., Strasdat, H., Kelly, P., & Davison, A. J. (2013a). Slam++: Simultaneous localisation and mapping at the level of objects. In *IEEE Conference on computer vision and pattern recognition (CVPR)* (pp. 1352–1359).
Salas-Moreno, R. F., Newcombe, R. A., Strasdat, H., Kelly, P. H., & Davison, A. J. (2013b). Slam++: Simultaneous localisation and mapping at the level of objects. In *IEEE Conference on computer vision and pattern recognition (CVPR)* (pp. 1352–1359).
Singh, A., Sha, J., Narayan, K. S., Achim, T., & Abbeel, P. (2014). Bigbird: A large-scale 3d database of object instances. In *2014 IEEE international conference on robotics and automation (ICRA), IEEE* (pp. 509–516).
Sünderhauf, N., Pham, T. T., Latif, Y., Milford, M., & Reid, I. (2017). Meaningful maps with object-oriented semantic mapping. In *IEEE/RSJ International conference on intelligent robots and systems (IROS), IEEE* (pp. 5079–5085).
Tchuiev, V., & Indelman, V. (2018). Inference over distribution of posterior class probabilities for reliable bayesian classification and object-level perception. *IEEE Robotics and Automation Letters (RA-L)*, *3*(4), 4329–4336.
Tchuiev, V., Feldman, Y., & Indelman, V. (2019). Data association aware semantic mapping and localization via a viewpoint-dependent classifier model. In *IEEE/RSJ International conference on intelligent robots and systems (IROS)*.
Teacy, W., Julier, S. J., De Nardi, R., Rogers, A., & Jennings, N. R. (2015). Observation modelling for vision-based target search by unmanned aerial vehicles. In *International conference on autonomous agents and multiagent systems (AAMAS)* (pp. 1607–1614).
Velez, J., Hemann, G., Huang, A. S., Posner, I., & Roy, N. (2012). Modelling observation correlations for active exploration and robust object detection. *Journal of Artificial Intelligence Research*, *44*, 423–425.

Yuri Feldman is a Ph.D. Computer Science student at the Autonomous Navigation and Perception Lab at the Technion, Israel Institute of Technology. He holds a B.Sc. degree in Computer Science, awarded by the Technion in 2010. His research focuses on semantic perception and mapping in uncertain environments.



Dr. Vadim Indelman is currently an Assistant Professor in the Department of Aerospace Engineering at the Technion, and he is also a member of the Technion Autonomous Systems Program (TASP). Upon joining the Technion in July 2014 Dr. Indelman established the Autonomous Navigation and Perception Laboratory (ANPL), where he and his research group investigate problems related to single and multi-robot collaborative autonomous navigation and perception, with a particular focus on online, accurate and reliable operation in uncertain and unknown environments. Prior to joining the Technion as a faculty member, Dr. Indelman was a postdoctoral fellow in the Institute of Robotics and Intelligent Machines (IRIM) at the Georgia Institute of Technology (between 2012 and 2014). He obtained his Ph.D. degree in Aerospace Engineering at the Technion - Israel Institute of Technology in 2011, and also holds B.A. and B.Sc. degrees in Computer Science and Aerospace Engineering, respectively, both awarded by the Technion in 2002.