
Deep Neural Network Uncertainty Estimation with Stochastic Inputs for Robust Aerial Navigation Policies

Fabio Arnez¹ Huascar Espinoza¹ Ansgar Radermacher¹ François Terrier¹

Abstract

Uncertainty quantification methods are required in autonomous systems that include deep learning (DL) components to assess the confidence of their estimations. However, to successfully deploy DL components in safety-critical autonomous systems, they should also handle uncertainty at the input rather than only at the output of the DL components. Considering a probability distribution in the input enables the propagation of uncertainty through different components to provide a representative measure of the overall system uncertainty. In this position paper, we propose a method to account for uncertainty at the input of Bayesian Deep Learning control policies for Aerial Navigation. Our early experiments show that the proposed method improves the robustness of the navigation policy in Out-of-Distribution (OoD) scenarios.

1. Introduction

Autonomous navigation in complex environments still represents a big challenge for autonomous systems. Particular instances of this problem are autonomous driving for self-driving cars and autonomous aerial navigation in the context of Unmanned Aerial Vehicles (UAVs). In both cases, the navigation task is addressed by first acquiring rich and complex raw sensory information (e.g., from cameras, radars, LiDARs, etc.), which is then processed to drive the robot towards its goal. Usually, this process is done in a modular fashion, where specific software components are linked together in the so-called *perception-planning-control* software pipeline (Siegwart et al., 2011; McAllister et al., 2017).

Modular pipeline components can be implemented using Deep Neural Networks (DNNs) or other non-learning methods (Grigorescu et al., 2020). DNNs have become a popular

choice thanks to their effectiveness in processing complex sensory inputs, and their powerful representation learning, that surpass the performance of traditional methods. Navigation policies implemented in a modular fashion admit a higher degree of interpretability and ease of analysis since components are developed in isolation. However, these architectures suffer from the accumulation of errors (coming from the contribution of each component’s erroneous outputs), which later harms the overall system performance (McAllister et al., 2017; Mueller et al., 2018). An alternative approach to modular pipelines is to map sensory inputs directly to control outputs using neural networks in an End-to-End (E2E) fashion (Grigorescu et al., 2020). This is an appealing paradigm where perception-planning-control blocks are trained jointly, often via imitation learning (Codevilla et al., 2018; 2019). Unfortunately, E2E training requires vast amounts of data (McAllister et al., 2017; Mueller et al., 2018) limiting its use to constrained scenarios.

Recently, some works from (Mueller et al., 2018; Bonatti et al., 2019; Wu et al., 2020) have explored the combination of modular and E2E learning approaches to get the benefits of both families. The goal of this hybrid approach is to learn an environment representation and a driving policy through dedicated DNNs. In this manner, an autonomous system can incorporate DL-based modules for perception and control. Learning intermediate (compact) representations enables the extraction of relevant environmental features that can be used later in downstream tasks (e.g. control policy training).

Despite substantial performance improvements introduced by DNNs, they still have significant shortcomings due to their opacity and specially their inability to represent confidence in their predictions. These downsides hinder the deployment of DL methods in safety-critical systems, where uncertainty estimates are highly relevant (Arnez et al., 2020; Loquercio et al., 2020). To overcome these limitations, the authors in (McAllister et al., 2017) propose the use of Bayesian Deep Learning (BDL) for implementing components used either in modular, E2E, or hybrid fashions. Bayesian methods in DL offer a principled framework to model data (*aleatoric*) and model (*epistemic*) uncertainties to represent the confidence in the outputs.

Nevertheless, system components that use BDL methods

¹Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France. Correspondence to: Fabio Arnez <fabio.arnez@cea.fr>.

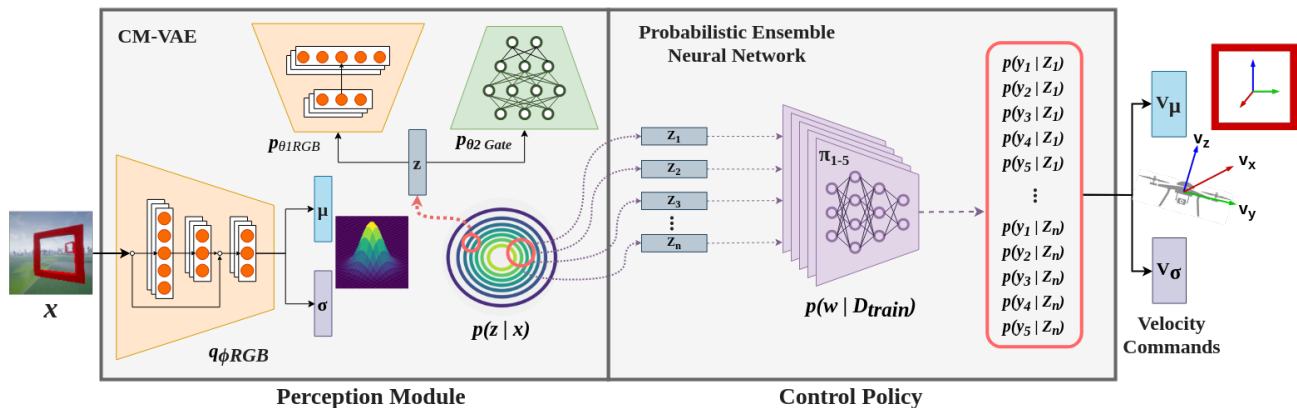


Figure 1. System Architecture for Autonomous Aerial Navigation.

should be able to admit uncertainty information as an input, to account for the uncertainty that derives from the outputs of preceding DL-based components. Considering a probability distribution in the input enables the propagation of uncertainty through different components to provide a representative measure of the overall system uncertainty. For example, erroneous or uncertain predictions of an object detector in the perception stage of the pipeline can result in an unexpected and unsafe motion plan from the trajectory planner (Ivanovic et al., 2021). In a similar fashion, in perception components that learn environmental compact representations (e.g. with Variational Autoencoders (Kingma & Welling, 2013)), the stochastic nature of the latent space can be used to model complex data-inherent uncertainty like multimodality (Depeweg et al., 2018). In the trajectory planner example, the stochastic latent variables from perception can help to predict multimodal motion plans in ambiguous scenarios like intersections.

In the literature, only few works such as (Henaff et al., 2018; McAllister et al., 2019; Fan et al., 2020) consider the use of uncertainty information from other DL components within a system. In this paper, we present a method to account for uncertainty information from stochastic variable inputs that are obtained from other DL components. This method represents a first effort towards handling uncertainty beyond perception in the software pipeline. We evaluate our approach in the context of autonomous aerial navigation, where we aim to learn a robust uncertainty-aware control policy that handles stochastic latent variables from a perception module. To the best of our knowledge, no previous methods handle continuous probability distributions at the inputs of a probabilistic neural network for control in an autonomous navigation task.

2. Methods

2.1. Autonomous Aerial Navigation with DNNs

In this work, we address the problem of autonomous aerial navigation, in which a UAV aims to traverse a set of gates (with unknown locations) in a drone racing context. To implement our method, we build on the work from (Bonatti et al., 2019). In this approach, the authors propose a navigation architecture that consists of two components implemented with DNNs: a perception module and a control policy. Figure 1 depicts the overall system architecture for autonomous navigation.

The objective of the perception module is to extract relevant information from the UAV environment and the current task. To do so, a low dimensional latent representation is learned using a deep generative model. A cross-modal variational autoencoder (CM-VAE) (Spurr et al., 2018) is an effective method to learn a rich, compact, and robust representation. CM-VAEs are variant of traditional VAEs that learn a single latent space with data from multiple sources (modalities). In this case-study, data modalities are represented by RGB images, and the pose of the next gate relative to the UAV body-frame. In the perception module, an RGB image x at the input is processed by the encoder $q_{\phi_{RGB}}$ into a multivariate normal distribution $\mathcal{N}(\mu, \sigma^2)$ from which latent vectors z are sampled. Later, each data modality can be recovered from the latent space using the decoder $p_{\theta1_{RGB}}$ and the estimator $p_{\theta2_{Gate}}$. For more details, we refer the reader to (Bonatti et al., 2019; Spurr et al., 2018).

In the control module, imitation learning is used to train a neural network control policy that maps latent stochastic variables (from CM-VAE) to UAV velocity commands. Different from (Bonatti et al., 2019), we train an ensemble of probabilistic control policies $\{\pi_i\}_{i=1}^M$ (see Figure 1) to account for both, model and data uncertainty. This can

Algorithm 1 Neural Network Predictive Distribution with Stochastic Inputs

Input: image \mathbf{x}^* , policy ensemble $\{\pi_{\mathbf{w}_i}\}_{i=1}^M \sim p(\mathbf{w}|\mathcal{D})$
Initialize: $\hat{\mu}_z, \hat{\sigma}_z, \hat{\mu}_\pi, \hat{\sigma}_\pi \leftarrow$ new empty Tensors
Sample: $\{\mathbf{z}_n\}_{n=1}^N \sim q_{\phi_{RGB}}(\mathbf{z}|\mathbf{x}^*)$.
for $n = 1$ **to** N **do**
 for $i = 1$ **to** M **do**
 $\hat{\mu}_\pi^{\mathbf{w}_i}, \hat{\sigma}_\pi^{\mathbf{w}_i} \leftarrow \pi_{\mathbf{w}_i}(z_n)$
 Insert $\hat{\mu}_\pi^{\mathbf{w}_i}$ into $\hat{\mu}_\pi$; Insert $\hat{\sigma}_\pi^{\mathbf{w}_i}$ into $\hat{\sigma}_\pi$
 end for
 $\hat{\mu}_{z_n} \leftarrow \text{mean}(\hat{\mu}_\pi)$
 $\hat{\sigma}_{z_n} \leftarrow \text{std}(\hat{\sigma}_\pi, \hat{\mu}_{z_n}, \hat{\mu}_\pi)$
 Insert $\hat{\mu}_{z_n}$ into $\hat{\mu}_z$; Insert $\hat{\sigma}_{z_n}$ into $\hat{\sigma}_z$
end for
 $\hat{\mu}_{\mathbf{y}^*} \leftarrow \text{mean}(\hat{\mu}_z)$
 $\hat{\sigma}_{\mathbf{y}^*} \leftarrow \text{std}(\hat{\sigma}_z, \hat{\mu}_{\mathbf{y}^*}, \hat{\mu}_\pi)$

be viewed as training a Bayesian Neural Network (BNN) (Gustafsson et al., 2019). In addition, our policy ensemble can handle uncertainty from the latent space (probability distribution) in its input to model complex uncertainty patterns, this is described in detail in the next section. To train the policy ensemble, we employ behavior cloning using raw images with velocity labels. We freeze all the weights from the perception encoder $q_{\phi_{RGB}}$ during the training process.

2.2. Uncertainty Estimation in DNNs with Stochastic Variable Inputs

In this work we propose a robust probabilistic control policy model that makes predictions about an output variable \mathbf{y} conditioned on an input \mathbf{x} , model parameters \mathbf{w} , and a latent variable $\mathbf{z} \sim p(\mathbf{z}|\mathbf{x})$. Our control policy model is then represented by the conditional probability distribution $p(\mathbf{y}|\mathbf{x}, \mathbf{z}, \mathbf{w})$. In our case-study, \mathbf{x} represents an RGB image, \mathbf{y} corresponds to UAV velocity commands, and \mathbf{z} represents the low dimensional latent space learned by the encoder model $q_{\phi_{RGB}}(\mathbf{z}|x)$.

To capture model uncertainty and handle stochastic inputs (probability distributions), we embrace the Bayesian approach to compute the distribution for a target variable \mathbf{y}^* associated with a new input \mathbf{x}^* , using the posterior predictive distribution:

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \int \int p(\mathbf{y}|\mathbf{z}, \mathbf{w})p(\mathbf{w}|\mathcal{D})p(\mathbf{z}|\mathbf{x}^*)d\mathbf{w}d\mathbf{z} \quad (1)$$

The above integral is intractable, and we rely on approximations to obtain an estimation of the predictive distribution. The distribution $p(\mathbf{w}|\mathcal{D})$ reflects the posterior over model weights given dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$, where \mathbf{X}, \mathbf{Y} represent the training inputs and \mathbf{Z} are the latent vectors observed

Table 1. Average Number of Gates Traversed by each Aerial Navigation Model (Perception and Control Policy DNNs).

MODEL	TRACK GATE RANDOM NOISE		
	RADIUS=0 HEIGHT=0	RADIUS=1 HEIGHT=2	RADIUS=1.5 HEIGHT=2.5
CM-VAE+BC	32	12	7
CM-VAE+BCE-UI1	32	16	10
CM-VAE+BCE-UI3	32	20	13
CM-VAE+BCE-UI5	32	20	17

during training. The posterior over the weights is difficult to evaluate, thus we can approximate the inner integral from Equation 1 using common BNNs methods (Arnez et al., 2020). In our case, each member of the ensemble can be viewed as a sample taken from the true posterior distribution over the weights (Gustafsson et al., 2019). Finally, the outer integral is approximated by taking n samples z_n from the latent space in our perception model. This last step can be seen as taking a better picture of the perception latent space (with multiple latent samples instead of a single one like in the baseline), leveraging in this way its stochastic nature to take robust control predictions. The overall method is illustrated in Figure 1, and summarized in Algorithm 1.

3. Early Experiments

For our experiments we used the aerial navigation architecture, the dataset, and the simulation environment from (Bonatti et al., 2019) as baseline. In this work, the architecture is composed by a CM-VAE, and a behavior cloning policy (CM-VAE+BC). To train both components, CM-VAE and BC policy, 300k and 17k RGB images (labeled) are used. In both cases the image size is 64x64 pixels.

We reproduced the baseline (CM-VAE+BC) and implement our method using Pytorch instead of Tensorflow. For the CM-VAE we used the same training settings as in the baseline for the unconstrained version. Different from the baseline, we implemented an ensemble of five neural-networks for our BC control policy (BCE for short). Each member of the ensemble outputs two values, corresponding to the predictive mean and variance, and is trained using the heteroscedastic loss function (Kendall & Gal, 2017; Lakshminarayanan et al., 2017). Our BCE policy is able to handle uncertainty at the input according to Algorithm 1, therefore is referred as **BCE-UI**. In consequence, our aerial navigation architecture for the experiments is referred as **CM-VAE+BCE-UIx**, where the last letter indicates the number of samples from the latent space to take into account for the policy predictions.

We evaluate our navigation architecture with the proposed method under controlled simulations that resemble the con-

ditions from the dataset collection. For this purpose, we create a circular track with eight equally spaced gates positioned initially in a radius of 8 m and constant height. To assess the robustness of the navigation policies, we generate new challenging tracks, adding random noise to each gate position (radius) and height. In this way, we force OoD operating conditions.

Table 1 shows the average number of gates traversed by each navigation policy under different noise levels for both gate radius (RN) and height (HN). We considered a maximum number of 32 gates which is equivalent to 4 laps in our track. Experimental results show that the navigation policy with our method outperforms the robustness of the other control policies (deterministic and with uncertainty representation only at the output). In fact, the version that includes more input samples is more robust to more drastic OoD conditions. In addition, we find that our navigation policy has a smooth behavior, while the baseline policy and the policy that represents uncertainty only at the output, present a turbulent behavior in noise tracks.

4. Conclusion

We presented a method to cope with stochastic variables at the input of BNNs used for control in an aerial navigation task. Based on our experiments, implementing a control policy that accounts for uncertainty at the input improves the robustness of the system in OoD scenarios, besides enabling a smooth behavior. The proposed method represents a step towards uncertainty handling beyond perception in the autonomous system software pipeline. Despite the benefits shown in the experiments, our method relies on sampling, which can be prohibitive in real-time systems. Another direction for future work would be the quality assessment of uncertainty estimates at component and system level, where we should consider the uncertainty propagation capability introduced by our method. This last point represents an important step towards deploying DL-components into safety-critical systems.

References

- Arnez, F., Espinoza, H., Radermacher, A., and Terrier, F. A comparison of uncertainty estimation approaches in deep learning components for autonomous vehicle applications. *Proceedings of the Workshop on Artificial Intelligence Safety 2020*, 2020.
- Bonatti, R., Madaan, R., Vineet, V., Scherer, S., and Kapoor, A. Learning visuomotor policies for aerial navigation using cross-modal representations. *arXiv preprint arXiv:1909.06993*, 2019.
- Codevilla, F., Müller, M., López, A., Koltun, V., and Dostovitskiy, A. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4693–4700. IEEE, 2018.
- Codevilla, F., Santana, E., López, A. M., and Gaidon, A. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9329–9338, 2019.
- Depeweg, S., Hernandez-Lobato, J.-M., Doshi-Velez, F., and Udluft, S. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning*, pp. 1184–1193. PMLR, 2018.
- Fan, T., Long, P., Liu, W., Pan, J., Yang, R., and Manocha, D. Learning resilient behaviors for navigation under uncertainty. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5299–5305. IEEE, 2020.
- Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.
- Gustafsson, F. K., Danelljan, M., and Schön, T. B. Evaluating scalable bayesian deep learning methods for robust computer vision. *arXiv preprint arXiv:1906.01620*, 2019.
- Henaff, M., Canziani, A., and LeCun, Y. Model-predictive policy learning with uncertainty regularization for driving in dense traffic. In *International Conference on Learning Representations*, 2018.
- Ivanovic, B., Lee, K.-H., Tokmakov, P., Wulfe, B., McAllister, R., Gaidon, A., and Pavone, M. Heterogeneous-agent trajectory forecasting incorporating class uncertainty. *arXiv preprint arXiv:2104.12446*, 2021.
- Kendall, A. and Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pp. 5574–5584, 2017.
- Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pp. 6402–6413, 2017.
- Loquercio, A., Segu, M., and Scaramuzza, D. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, 5(2):3153–3160, 2020.

- McAllister, R., Gal, Y., Kendall, A., Van Der Wilk, M., Shah, A., Cipolla, R., and Weller, A. In *Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning*. International Joint Conferences on Artificial Intelligence, Inc., 2017.
- McAllister, R., Kahn, G., Clune, J., and Levine, S. Robustness to out-of-distribution inputs via task-aware generative uncertainty. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2083–2089. IEEE, 2019.
- Mueller, M., Dosovitskiy, A., Ghanem, B., and Koltun, V. Driving policy transfer via modularity and abstraction. In *Conference on Robot Learning*, pp. 1–15. PMLR, 2018.
- Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D. *Introduction to autonomous mobile robots*. MIT press, 2011.
- Spurr, A., Song, J., Park, S., and Hilliges, O. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 89–98, 2018.
- Wu, Q., Gong, X., Xu, K., Manocha, D., Dong, J., and Wang, J. Towards target-driven visual navigation in indoor scenes via generative imitation learning. *IEEE Robotics and Automation Letters*, 6(1):175–182, 2020.