Efficient Multi-Class Out-of-Distribution Reasoning for Perception Based Networks: Work-in-Progress

Shreyas Ramakrishna*§, Zahra Rahiminasab†§, Arvind Easwaran †, Abhishek Dubey*
* *Vanderbilt University*
† *Nanyang Technological University*

***Abstract*—Perception-based deep neural networks used in Cyber Physical Systems are known to fail when faced with inputs that are out-of-distribution (OOD). OOD detection is a complex problem as we need to first identify the shift in the test data from the training distribution and then we need to isolate the responsible generative factor(s) (weather, lighting levels, traffic density, etc.). Unlike the state of the art that uses multi-chained one-class classifiers, we propose an efficient single monitor that uses the principle of disentanglement to train the latent space of a variational autoencoder to be sensitive to distribution shifts in different generative factors. We demonstrate our approach using an end-to-end driving controller in the CARLA simulator.**

***Index Terms*—β-VAE, Disentanglement, Inductive Conformal Prediction, Mutual Information Gap.**

I. INTRODUCTION

Perception-based deep neural networks (DNN) are being used in the automotive Cyber-Physical Systems (CPS) to perform tasks like object detection, object classification, and end-to-end actuation control, etc. They are trained with large datasets of images with multimodal generative factors like time-of-day (day, night), weather (cloudy, slight-rain), and traffic-density (no-traffic, traffic), etc. Despite the exceptional performance, fatal incidents like Tesla's autopilot crash [1] and Uber self-driving car crash [2] have shown these components to fail, when the operational input is out-of-distribution (OOD). *For the safety of CPS, it is essential to detect OOD images and find the responsible generative factors*. At this point, the problem becomes multi-label OOD detection.

State of the art OOD detection in multi-label datasets is addressed [3] by synthesizing the dataset into partitions based on labels, and then train a one-class classifier for each partition. One-class classifiers like Deep SVDD [4] and Variational Autoencoder (VAE) [5] are widely used to detect OOD's. Especially, methods using the concept of VAE based reconstruction error [6] have become popular. However, these methods have problems. First, the reconstruction based methods are known to be less robust to errors [7]. Second, these multi-chain classifiers are computationally expensive - images from automotive datasets like nuScenes [8] have greater than 10 generative factors, and using one classifier for each factor would be computationally expensive for automotive CPS. The third problem is the issue of approximate partitions. Synthesizing the autonomous datasets into clear partitions (with mutually exclusive partitions) is difficult because the generative factors are not independent. This hinders directly using existing classifier techniques to detect OOD. Finally, most of the existing methods rely on point predictions; that is, they do not use the time series of input images, which is required for robust detection.

Recently, there has been a growing interest in using the latent space learned by a VAE for OOD detection [7]. At the same time, there has been progress in learning disentangled representations [5] in the latent space. Disentanglement means that each latent unit mainly encodes probability distribution related to a specific generative factor. However, achieving the disentanglement of the latent space is a hard problem, as it requires the generative factors to be independent. However, images obtained from real-world datasets, including simulators, do not have this property, thus making disentanglement a challenging objective to achieve.

Our contribution in this paper is to describe an approach to train a VAE with partial disentanglement and use it for OOD detection and reasoning in images. For this, we use a form of VAE called the β-VAE. An appropriate β (> 1) and the right size of the latent space (n) is required to achieve disentanglement. For finding the right hyperparameter combination, we use a heuristic based on an information-theoretic metric called the Mutual Information Gap (MIG) [9]. After selecting the β-VAE, we apply the Inductive Conformal Prediction (ICP) [10] framework using a KL-divergence (computed with respect to the distributions generated by the latent space variables) based non-conformity score that indicates how similar the input image distributions are to the training set distributions. The non-conformity scores are then used to compute a p-value, which is used to compute the mixture martingale [11] over a window M of image sequences to improve the detector robustness.

II. OOD MONITORING APPROACH

We use the latent space generated by a β-VAE along with the KL-divergence metric to perform OOD detection and reasoning. A β (=1) imposes a bottleneck on this information flow, thus making it uninformative and focuses primarily on the reconstruction of images. However, an appropriate β (> 1) can help the latent space learn the distribution of the generative factors better. Eventually, at a higher β, the latent space variables become sensitive to different generative factors. Also, for an appropriate combination of β (> 1) and latent space size (n), the latent space gets disentangled.

§These Authors have equally contributed

Exhaustively searching for these hyperparameters is expensive. So, we devise a two-step heuristic approach that includes a simple random search [12] to select the best β value for each n, which results in the lowest evidential lower bound (ELBO) [5]. For each $n \in [30, 40, 50, 100, 200, 300, 400, 500, 1000]$ we find the $\beta \in [1, 1.1, 1.2, 1.3, 1.4, 1.5, 2, 3, 4, 5, 10]$ which results in a minimal ELBO over 100 iterations. We then select and train 10 different β-VAE's from the shortlisted [n,β] combination. Next, to shortlist one β-VAE, which generates a well disentangled latent space, we use an information theory metric called Mutual Information Gap (MIG) [9]. MIG is a metric based on information theory that averages the difference between the empirical mutual information of the two most informative latent units for each image factor and normalizes this result by the entropy of the factor. We believe the MIG is higher for a better disentangled latent space, so we select a β-VAE that results in the highest MIG.

Next, given a β-VAE with the latent unit set $\mathcal{L}$, we select a detector, which is a latent unit subset $\mathcal{L}_d \subseteq \mathcal{L}$ that can detect distribution shifts from the training dataset. Further, we select reasoners, which are latent unit(s) $\mathcal{L}_f \subseteq \mathcal{L}_d$ that can identify distribution shifts for each generative factor. For identifying these latent units, we perform a latent unit comparison method, similar to the one in [5]. The key idea (as in [5]) is that informative latent units have higher KL-divergence from the normal distribution $\mathcal{N}(0, 1)$, while the uninformative ones will have KL-divergence close to zero.

At runtime, an operational test image is passed to the encoder of the β-VAE monitor to generate the latent unit distributions. The latent units ($\mathcal{L}_d$ and $\mathcal{L}_f$), are plugged into the Inductive Conformal Prediction [10] framework and non-conformity scores are computed using KL-divergence. Using this score, a p-value is computed, which is a fraction of the calibration observations that have a non-conformity measure above the test observation. Then, the past M p-values are used to compute the mixture martingale [11] at time t. The martingale will grow over time only if there are consistently low p-values within the time window $[t - M + 1, t]$. A consistently growing martingale indicates the test images are OOD. Finally, a cumulative sum (CUSUM) is computed over the martingale value as follows: S_0 = 0 and S_{t+1} = max(0, S_t+M_{t-1}-ω), where ω is the weight assigned to prevent S_t from consistently increasing to a large value. Then, the S_t value is compared against a threshold (τ) to detect OOD and identify feature variations.

III. Experimental Results

We applied our approach to an autonomous end-to-end driving controller built in the CARLA simulator [13]. The block diagram of the software components for the end-to-end control of an autonomous vehicle in CARLA is shown in Fig. 1. The components marked in green come pre-built with the simulator, and the ones in blue have been designed by us. These components are implemented as separate python processes, and they communicate using the ZMQ [14] publish-subscribe communication pattern. For autonomous control of the vehicle, an NVIDIA DAVE-II [15] Learning Enabled Component (LEC) is used to predict steering values using the front-facing camera images.

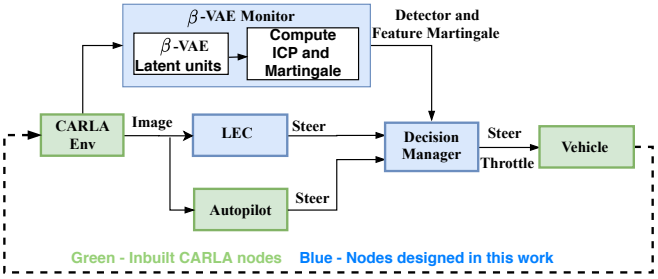

Fig. 1: A block diagram of the components for end-to-end control of an autonomous vehicle using a LEC in the CARLA simulator. The components in green come pre-built with CARLA, while the components in blue are designed for our experiments. Also, all the CARLA components require a GPU for the simulation engine, while all the other components are run in a resource limited setting.

To train the control LEC and β-VAE monitor, we collected 4780 camera images using CARLA's inbuilt autopilot mode. These images belonged to 3 generative factor partitions of rain intensity (no-rain, mild-rain, and heavy-rain), illumination levels (low, medium, high), and time-of-day (day, evening). Using the monitor design steps (Section II), we selected a β-VAE with β = 1.4 and n = 30 which resulted in the highest MIG (0.000072). Then, we identified a subset of 9 latent units that encoded most information about the training image representations and used it as $\mathcal{L}_d$.

Further, we identified that rain-intensity and illumination-level had an impact on the steering predictions of the LEC, so we tried to identify the latent units encoding their information. From $\mathcal{L}_d$ we identified 1 latent unit each that encoded only information about rain-intensity (L_{30}), and illumination-level (L_3). Besides, we empirically found the thresholds for the CUSUM detectors. For the detector CUSUM, we selected $\omega = 14$ and $\tau = 100$. These parameters resulted in the lowest false positives ($< 1\%$) when tested for 6 different in-distribution and OOD scenes. For the feature detector CUSUM, we selected $\omega = 18$ and $\tau = 130$. These parameters were found empirically. Finally, we selected a sliding window size of $M = 20$ for computing the martingale.

Using the identified parameter values, we evaluated the performance of the monitor with 4 experiments (See Fig. 2). The first experiment was an in-distribution scenario, so the martingales of the detector and reasoners remain below the threshold. In experiment 2, the rain intensity changes at $t = 3s$, and only the martingale corresponding to that specific factor, and the detector martingale increases after $t = 3s$ (Similar results on changing illumination level is not shown due to lack of space). In experiment 3, both the rain intensity and illumination level is changed to OOD at $t = 3s$. The martingales of generative factors, as well as the detector increase after $t = 3s$. Finally, experiment 4 was designed to show the robustness of our approach to an unseen scene with a new road segment, but with in-distribution levels of rain intensity and illumination

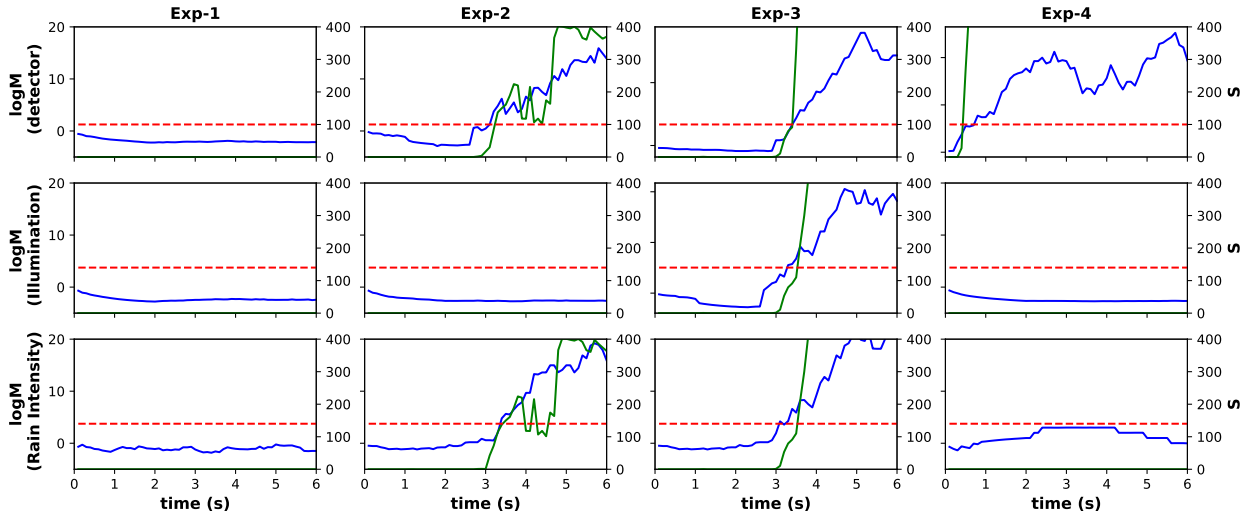

Fig. 2: Performance of the β-VAE monitor for experiments 1 - 4, discussed in Section III. The blue lines represent the martingale values, the solid green lines represent the CUSUM (S) values, and the dotted red lines represent the threshold (τ) for CUSUM comparison.

levels. In this case, the martingales corresponding to the rain intensity and illumination levels do not increase above the thresholds. However, because of considerable variation in the generative factors when compared to the training set, the detector martingale starts to increase from $t = 0s$. For these experiments, the detector had precision and recall of 97.2% and 86.32%, and an F1-score of 91.10%. Also, the average detection time using all 30 latent units was 88 ms, and this was reduced to 74.09 ms when using the selected 9 latent units. This 16% time reduction in the detection time, improved the simulation Frame Per Seconds (FPS) from 11 to 14. Further, the reasoners took an average time of 48.7 ms.

IV. Conclusion and Future Work

We proposed a latent space-based β-VAE monitor for OOD detection and reasoning. For this, we described a heuristic-based method to design the β-VAE monitor and to select the latent units encoding information about generative factors of interest (e.g. illumination level, rain-intensity). Finally, we used the selected latent units in the ICP framework for run-time monitoring. Our evaluations using the CARLA simulator shows the monitor to reliably detect OOD with an F1-score of 91.10% in a short time of 74 ms.

This work-in-progress uses a heuristic-based method using random search and MIG to select the β-VAE hyperparameters, and we are currently working on a principled mechanism for selecting these hyperparameters. We are also working towards comparing our OOD detection approach to other One-class classifier techniques in the literature.

Acknowledgement This work was supported in part by the DARPA Assured Autonomy project, Air Force Research Laboratory and partially under the MoE, Singapore, Tier-2 grant #E2019-T2-2-040.

References

[1] B. Vlasic and N. E. Boudette, "'self-driving tesla was involved in fatal crash,'us says," *New York Times*, vol. 302016, 2016.

[2] P. Kohli and A. Chadha, "Enabling pedestrian safety using computer vision techniques: A case study of the 2018 uber inc. self-driving car crash," in *Future of Information and Communication Conference*. Springer, 2019, pp. 261–279.

[3] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine learning*, vol. 85, no. 3, p. 333, 2011.

[4] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*, 2018, pp. 4393–4402.

[5] I. Higgins, L. Matthey, A. Pal, C. Burgess, X.-a. Glorot, M. Botvinick, S. Mohamed, and A.-d. Lerchner, "Beta-VAE: Learning basic visual concepts with a constrained variational framework." *ICLR17*, 2016.

[6] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, no. 1, 2015.

[7] T. Denouden, R. Salay, K. Czarnecki, V. Abdelzad, B. Phan, and S. Vernekar, "Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance," *arXiv preprint arXiv:1812.02765*, 2018.

[8] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[9] T. Q. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," in *Advances in Neural Information Processing Systems*, 2018, pp. 2610–2620.

[10] G. Shafer and V. Vovk, "A tutorial on conformal prediction," *Journal of Machine Learning Research*, vol. 9, no. Mar, pp. 371–421, 2008.

[11] V. Fedorova, A. Gammerman, I. Nouretdinov, and V. Vovk, "Plug-in martingales for testing exchangeability on-line," *arXiv preprint arXiv:1204.3251*, 2012.

[12] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in neural information processing systems*, 2011, pp. 2546–2554.

[13] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," *arXiv:1711.03938*, 2017.

[14] P. Hintjens, *ZeroMQ: messaging for many applications.* " O'Reilly Media, Inc.", 2013.

[15] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.