Appendix

A. *Network Architectures*

Table I shows the network architecture for the driving policy described in the paper. The architecture contains three convolutional layers and two dense layers. The input shape is $[84 \times 84 \times 4]$ (four consecutive grayscale images). The output is a vector of probabilities with a shape equal to the number of possible actions.

Table II shows the network architecture for the affect-based reward function described in the paper. The architecture contains three convolutional layers and two dense layers. Batch normalization is applied prior to each intermediate layer. The input shape is $[84 \times 84 \times 3]$ and contains a single RGB image. The output is a vector of two values. The first is the positive affect-based value that is being used by our affect-based policy, and the second is the proposed reward output from [1] which is being examined using our model architecture in the fourth baseline described in the experiments.

Layer	Act.	Out. shape	Parameters
Conv2D	ReLU	$16 \times 8 \times 8$	4.1k
Conv2D	ReLU	$32 \times 4 \times 4$	8.2k
Conv2D	ReLU	$32 \times 3 \times 3$	9.2k
Dense	ReLU	256	401k
Dense	Softmax	5	1.2k
Total			424k

TABLE I: CNN architecture for the navigation policy.

Layer	Act.	Out. shape	Parameters
Conv2D	ReLU	$32 \times 5 \times 5$	2.4k
Conv2D	ReLU	$48 \times 4 \times 4$	24.6k
Conv2D	ReLU	$64 \times 4 \times 4$	49.2k
Dense	ReLU	2048	8.4M
Dense	Linear	2	4k
Total			8.4M

TABLE II: CNN architecture for the affect-based reward function.

Table III shows the network architecture for the VAE model described in the paper. The encoder and the decoder composed of five layers each, with batch normalization prior to each intermediate layer. The input shape is $[64 \times 64 \times 3]$ and contains a single RGB image. The output of the encoder is an 8-dimensional latent space representation. The output shape of the decoder is $[64 \times 64 \times 3]$ for a single RGB/segmentation image, or $[64 \times 64 \times 1]$ for a depth estimation map.

For more details about the training procedure and parameters, the code is available in this repository.

Layer	Act.	Out. shape	Params
Encoding Layers			
Conv2D	ReLU	$64 \times 4 \times 4$	3.1k
Conv2D	ReLU	$128 \times 4 \times 4$	131k
Conv2D	ReLU	$256 \times 4 \times 4$	524k
Dense	ReLU	1024	9.4M
Dense	Linear	16	16.4k
Decoding Layers			
Dense	ReLU	1024	9.2k
Dense	ReLU	6272	6.4M
Conv2D Transpose	ReLU	$128 \times 4 \times 4$	262k
Conv2D Transpose	ReLU	$64 \times 4 \times 4$	131k
Conv2D Transpose	Sigmoid	$3 \times 4 \times 4$	3k
Total			16.9M

TABLE III: Architecture for the convolutional VAE model.

B. *Reward Multiplication Factor*

Our method relies on adding the reward component on top of the action probabilities such that it will maximize the exploration results. To find the best γ, we performed the experiment shown in the coverge table that is in the paper for a range of potential values, and then we used this γ to perform the rest of the experiments. An example of an experiment to find the γ can be seen in Fig. 1.

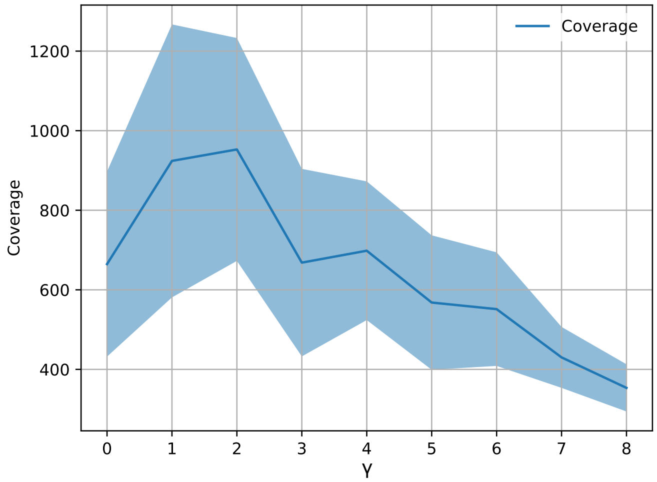

Fig. 1: Mean coverage as a function of gamma when searching for the right gamma for IL + [1]. The plot shows that $\gamma = 2$ resulted in the highest coverage.

References

[1] D. McDuff and A. Kapoor, "Visceral machines: Risk-aversion in reinforcement learning with intrinsic physiological rewards," *International Conference on Learning Representations (ICLR)*, 2019.