JOCA: Task-Driven Joint Optimisation of Camera Hardware and Adaptive Camera Control Algorithms

Supplementary Material

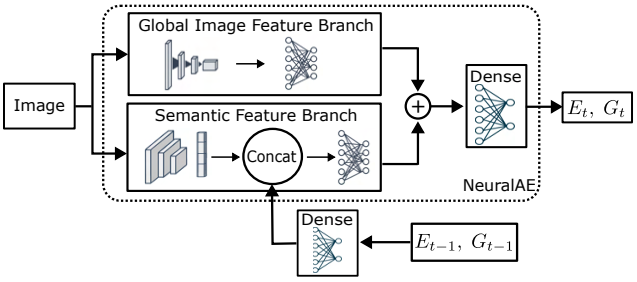

Figure 1. Adaptive camera control algorithm. We adopt the architecture from NeuralAE [7] as the main architecture for the ACC algorithm. We modify NeuralAE by using a single camera instead of two cameras, and by concatenating features from the predicted dynamic parameters at the previous step to the extracted features in the semantic feature branch for temporal consistency. Finally, we allow it to predict both exposure time and gain, rather than a single exposure value as in its original version. Figure is adapted and modified from [7].

1. Architecture of Adaptive Camera Control Model

We adopt and modify the NeuralAE model from [7] as the adaptive camera control (ACC) algorithm. An illustration of the model architecture is shown in Fig. 1.

The model retains both the global image feature branch and the semantic feature branch from the original implementation. The global image feature branch uses a 3-layer 1D convolutional neural network to extract features from multi-scale histograms of the images, followed by two linear layers to obtain dense features.

The semantic feature branch reuses the ResNet [4] feature extractor from the downstream tasks, taking the conv2 layer output as input. This feature is compressed with a 2D convolution and split into three compressed feature maps (CFMs). Average pooling is applied to the first CFM, while max pooling is applied to the last two. The original compressed feature (before splitting) also undergoes average pooling. These pooled features from the CFMs and the compressed feature are concatenated. We modify this step by also concatenating a 32-dimensional feature map derived from the dynamic parameters predicted at the previous timestep. This feature map is obtained by expanding the two input parameters through two linear layers. The concatenated feature then passes through two linear layers to produce a dense feature of the same size as the global image feature branch output. To accommodate the additional features, the input size of these linear layers is increased by 32 (original: 784, modified: 816).

Finally, the outputs of the two branches are summed, and a linear layer is applied to reduce the summed feature to match the number of required dynamic parameters. Unlike the original implementation in [7], which outputs only a single exposure value, our modified model predicts both exposure time and gain.

Further architectural details can be found in [7]. We maintain the same number of parameters for each layer as in the original, with the only modifications being the concatenation of previous dynamic parameter features and the expanded output size.

2. Details on Optimisation

We apply a genetic algorithm (GA) [5] to optimise the camera hardware, and use the proposed DF-Grad method to train the ACC algorithm. The implementations of these methods are detailed in this section.

2.1. Camera Hardware

We use a population size of 10 over 35 generations, with 50% of the population generating offspring through crossover and mutation. A uniform crossover process is applied, where each parameter of the offspring is randomly selected from one of the parents. The mutation process multiplies each parameter by a random factor between 0.8 and 1.2. Initial camera parameters are randomly sampled from their respective feasible ranges. These hyperparameters and implementation choices follow the recommendations of [9], which optimises a similar number of parameters in its experiments.

2.2. Adaptive Camera Control Algorithm

We determine the weights of GA loss and task loss when updating the ACC model, the population size used to optimise perturbations for dynamic parameters, and the frequency of applying the GA loss based on experimental results. Preliminary experiments are first conducted to identify suitable value ranges for these hyperparameters. We then select the final values through empirical tuning based on performance. The results and analysis from the empirical value selection in the chosen ranges are presented in this section.

Weight of Loss Function Our DF-Grad optimisation scheme uses a combined task loss and GA loss, as described in Sec. 5 of the main paper, to train the ACC algorithm. We

Table 1. Camera designs and object detection performance for task loss weights of 5, 7, and 9 used during ACC algorithm training. The results indicate that a weight of 7 yields the best task performance.

Weight	Camera Parameters					Performance	
	Forward Position x (m)	Height z (m)	Focal Length f (mm)	Sensor Size $w \times h$ (mm)	Pixel Size p (μm)	Object Detect. mAP ↑	True Positive Ratio TP/All@180° ↑
5	1.34	1.35	4.91	5.70×4.28	2.2	0.293	0.366
7 (Proposed)	1.35	1.3	5.19	7.37×4.92	2.4	**0.325**	**0.390**
9	1.31	1.30	5.76	6.77×5.66	2.74	0.296	0.361

Table 2. Camera designs and object detection performance for population sizes of 3, 4, and 5 used in GA optimisation of dynamic parameter perturbations. The results show that a population size of 4 achieves the best task performance.

Population Size	Camera Parameters					Performance	
	Forward Position x (m)	Height z (m)	Focal Length f (mm)	Sensor Size $w \times h$ (mm)	Pixel Size p (μm)	Object Detect. mAP ↑	True Positive Ratio TP/All@180° ↑
3	2.2	1.57	6.09	8.66×4.34	2.25	0.296	0.388
4 (Proposed)	1.35	1.3	5.19	7.37×4.92	2.4	**0.325**	**0.390**
5	1.80	1.70	9.81	14.13×7.45	3.45	0.306	0.381

Table 3. Camera designs and object detection performance when updating the ACC algorithm with GA loss every 3, 4, and 5 steps. The results show that an update frequency of 4 achieves the best task performance.

Update Frequency	Camera Parameters					Performance	
	Forward Position x (m)	Height z (m)	Focal Length f (mm)	Sensor Size $w \times h$ (mm)	Pixel Size p (μm)	Object Detect. mAP ↑	True Positive Ratio TP/All@180° ↑
3	1.70	1.70	5.94	8.66×4.34	2.25	0.301	0.379
4 (Proposed)	1.35	1.3	5.19	7.37×4.92	2.4	**0.325**	**0.390**
5	2.01	1.70	4.51	5.70×4.28	2.2	0.320	0.381

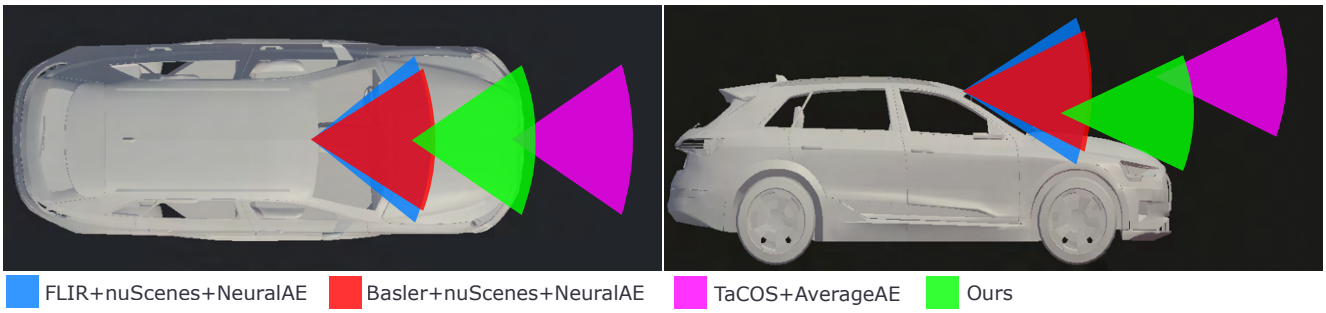

Figure 2. Visualisation of the camera FoVs and placements for our designed camera, the FLIR/Basler cameras using the nuScenes placement, and the camera designed by TaCOS with the AverageAE method.

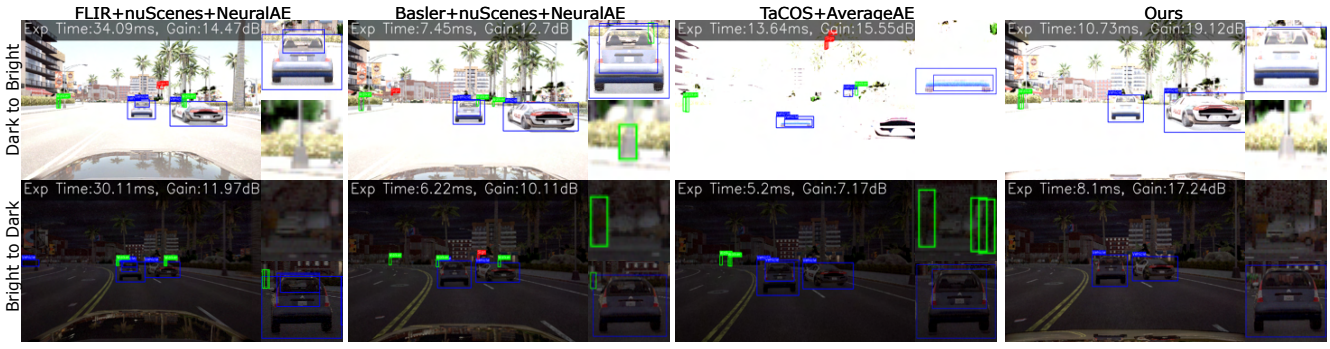

Figure 3. Qualitative results under abrupt illumination change from dark to bright and vice versa. The results shows an increased detection rate with the NeuralAE method, highlighting the advantages of using a learning-based ACC algorithm in handling such transitions.

Figure 4. Additional qualitative results on synthetic data comparing our proposed approach with baselines. The results demonstrate that our method more accurately detects small and distant objects, particularly in challenging scenarios with increased noise and motion blur.

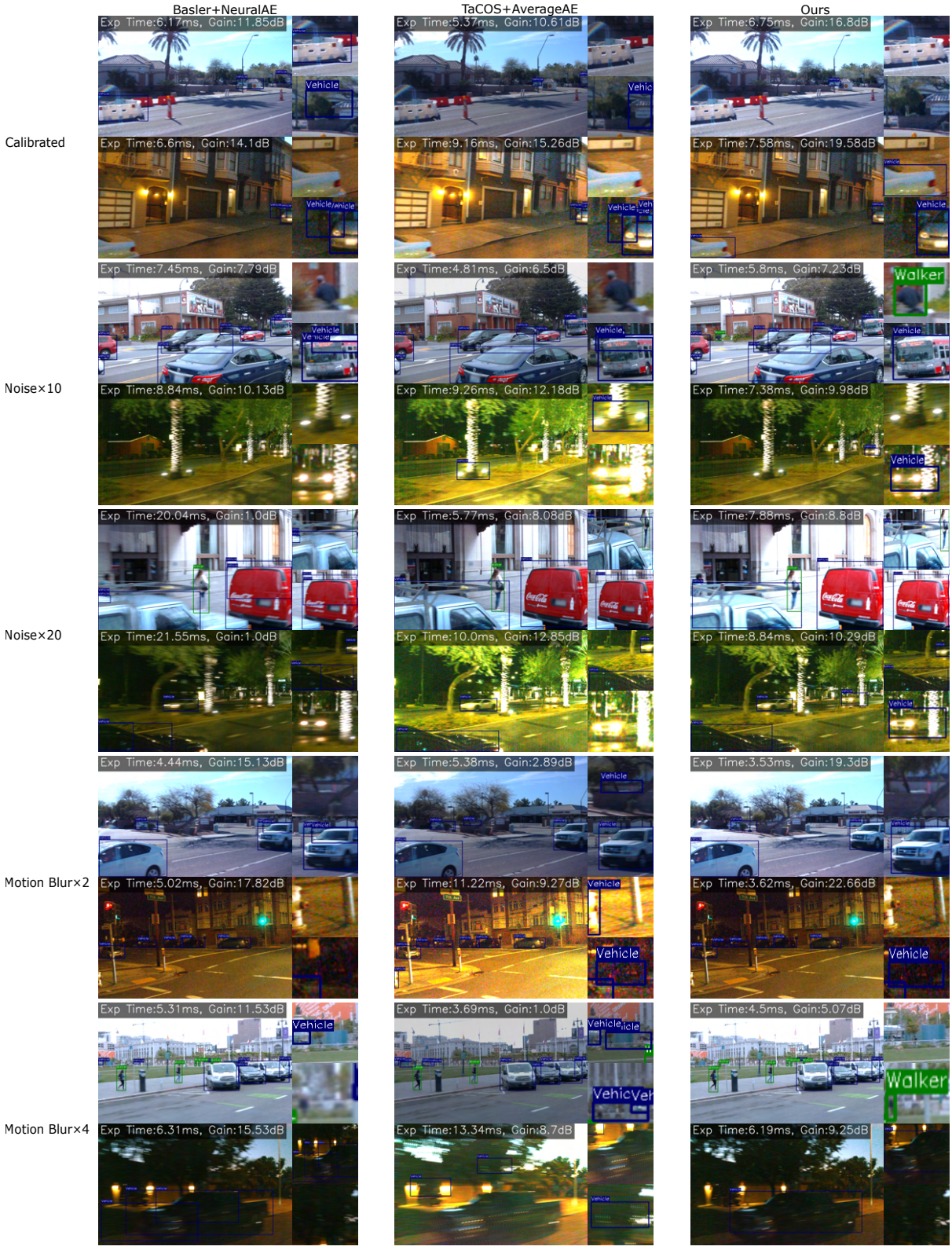

Figure 5. Qualitative results on real-world images show that our method consistently outperforms all baselines, further validating its practicality. Competing methods exhibit noticeable false positives, such as the examples on rows 1 and 10, and other examples show a mix of both false positives and false negatives from the baselines.

fix the weight of the GA loss at 1, and vary the task loss weight to determine the optimal value. We show the results of using a weight of 5, 7, and 9 in Tab. 1. The results suggest that a weight of 7 best balances the unsupervised learning signal from the task model with the supervision from GA, yielding the best task performance. The results also indicate that the camera hardware design is relatively insensitive to changes in the task loss weight.

Population Size We conduct experiments with varying population sizes for the GA optimisation of exposure perturbations. Results are summarised in Tab. 2. A population size of 4 provides the best performance. Smaller population sizes cause the optimiser to converge to local optima due to reduced diversity, while larger population sizes increase convergence time [9], potentially requiring a longer update frequency and thus lengthier overall optimisation.

Update Frequency As discussed in the main paper, the GA loss is used to supervise the ACC network every i steps. The update frequency i represents the number of steps allowed for GA to optimise perturbations before applying the supervision. Since illumination conditions can change frequently, applying the GA loss intermittently allows new perturbations to be optimised for each lighting condition. We show the results of using an update frequencies of 3, 4, and 5 in Tab. 3. The results indicate that applying GA loss every 4 steps offers a good balance, giving the GA sufficient time to adapt to changing environments while providing consistent supervisory signals to the ACC algorithm.

We note that population size and update frequency are closely related. A larger population size generally requires a larger value of update frequency, as it increases convergence time [9]. In our experiments, we first select a population size based on optimisation time and early performance with an intuitive update frequency, then refine the update frequency through empirical testing.

GA Implementation We optimise the perturbations of dynamic parameters using a GA. Following [9], 50% of the population generates offspring via uniform crossover. For mutation, we apply a random multiplication factor between 0.9 and 1.1 to each parameter. This tighter range avoids abrupt changes in dynamic settings, which could lead to undesirable visual artefacts such as flickering.

3. Additional Results on Camera Design

In this section, we present additional results from our camera design experiments. These include illustrations of the cameras' fields of view (FoV) and placements, performance under rapid illumination changes, and additional qualitative comparisons.

3.1. Camera FoV and Placement

We illustrate the FoVs and placements of the camera designed by our proposed method, the FLIR [3] and Basler [1] cameras (using the nuScenes placement [2]), and the camera designed by TaCOS [9] with the AverageAE [6] method in Fig. 2. These visualisations correspond to the calibrated design scenario.

From the figure, we observe that both our method and TaCOS produce camera designs with moderate FoVs, striking a balance between maximising mAP and maintaining a high true positive (TP) detection ratio. The camera placements selected by our method also provide clear views of the environment for accurate perception.

3.2. Abrupt Illumination Change

To simulate real-world situations with abrupt illumination changes, such as entering or exiting tunnels, we include a scenario in which the illumination changes abruptly between day and night.

The performance of different methods in this setting is shown in Fig. 3. Our findings show that methods employing learning-based ACC algorithms predict moderate dynamic parameters, preventing saturation when transitioning from dark to bright scenes. In contrast, the AverageAE method relies solely on the previous frame's image intensity, often resulting in saturated images during such abrupt changes. Performance for all methods under transitions from bright to dark is comparable to that in constant dark conditions.

3.3. Additional Qualitative Results on Synthetic Data

We provide further qualitative comparisons between our method and the baselines across all design scenarios using the synthetic images in Fig. 4. As discussed in the main paper, our method enables the ACC algorithm to account for non-differentiable effects such as motion blur, which baseline methods fail to model. Specifically, our method predicts lower gain values in high-noise scenarios to improve signal-to-noise ratio (SNR), and it predicts shorter exposure times in conditions with increased motion blur to reduce image degradation. In addition, our method co-designs camera hardware, such as pixel size, to complement these adaptive settings. Larger pixel sizes are chosen when higher noise and motion blur are expected, to increase the number of collected photons for higher SNRs. These design choices allow our method to detect small and distant objects more reliably across both calibrated and challenging scenarios.

3.4. Qualitative Results on Real-World Data

Fig. 5 presents qualitative comparisons between our method and the baselines across varying scenarios in the Waymo Open dataset [8]. The results show consistent

trends with the synthetic experiments and validate the practicality of our method. Our method effectively balances motion blur and SNR, achieving improved task performance across all scenarios, particularly under challenging conditions, owing to the proposed DF-Grad method. Moreover, by jointly optimising camera hardware, our method attains higher SNRs while maintaining sufficient resolution for object detection by changing the pixel size, as reported in the main paper.

References

[1] *Basler dart daA1280-54uc*. Basler AG, 2024. Rev. 85. 5

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 5

[3] *FLIR FLEA®3 USB3 Vision*. FLIR Integrated Imaging Solutions Inc., 2017. Rev. 8.1. 5

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

[5] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992. 1

[6] ARM Ltd. Mali-c71ae: Advanced ISP for automotive and industrial, 2020. 5

[7] Emmanuel Onzon, Fahim Mannan, and Felix Heide. Neural auto-exposure for high-dynamic range object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7710–7720, 2021. 1

[8] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 5

[9] Chengyang Yan and Donald G. Dansereau. TaCOS: Task-specific camera optimization with simulation. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 2052–2062, 2025. 1, 5