DUALAD: Disentangling the Dynamic and Static World for End-to-End Driving

Supplementary Material

In this supplementary document, we first provide implementation details of our proposed approach. Furthermore, we present additional evaluation metrics for all perception tasks tackled by DUALAD. Next, we discuss experimental findings regarding our design choices and temporal consistency. Finally, we provide a detailed runtime analysis for different variants of our model and show additional qualitative results in the attached video file.

6. Implementation Details

Our work is built using the MMDetection3D framework[4]. Furthermore, we inherit various design choices from StreamPETR [26, 30], UniAD [10, 27] and VAD [11, 28]. We truly thank all authors and contributors of those projects. Our main model configuration closely follows StreamPETR [26, 30] since our dynamic stream design inherits the proposed query propagation through time as well as the geometric positional encodings for object-to-image cross-attention. All choices for the static stream are adopted from UniAD [10].

Data Augmentation: We use the six surround camera images of nuScenes as input, down scaled to a resolution of 800×320 pixels. During training, we apply a random crop augmentation by choosing a random crop of $47\,\% - 62.5\,\%$ of the image before down scaling.

Model Settings: We use a VovNet-V2-99 [12] as image backbone and use the last two feature scales as input to the FPN [17]. As in previous work, a latent dimension $L = 256$ is adopted for all latent embeddings of our model. We use $|\mathcal{Q}_{\text{obj}}| = 900$ object queries consisting of the top-k propagated from the previous time step with $k = 256$ and 644 newly spawned objects queries respectively. For the BEV-queries we follow UniAD [10] and use $|\mathcal{Q}_{\text{BEV}}| = 200 \times 200$. The used detection range is $[-51.2\,\text{m}, 51.2\,\text{m}]$ for x and y direction, resulting in an effective grid resolution of $0.512\,\text{m}$.

The proposed dual-stream transformer utilizes six consecutive layers and performs self-attention within $\mathcal{Q}_{\text{obj}}$, cross-attention of $\mathcal{Q}_{\text{obj}}$, temporal self-attention of $\mathcal{Q}_{\text{BEV}}$ and the interpolated grid queries from the last frame [10, 14], cross-attention from $\mathcal{Q}_{\text{BEV}}$ to image features as in [14] and dynamic-static cross-attention of $\mathcal{Q}_{\text{obj}}$ and $\mathcal{Q}_{\text{BEV}}$. For the dynamic object cross-attention to the image features we only choose the highest spatial resolution feature scale as in [26, 30].

During training, we adopt query-denoising [13] and streaming video training as proposed in [30] to accelerate the convergence as well as Flash-Attention [5] to reduce the memory requirements. With the aforementioned settings, the training for 24 epochs requires 18 GB of GPU memory and takes approximately one day for stage-I and two days for stage-2 on eight NVIDIA A100 GPUs.

7. Performance Evaluation

We provide evaluation results for various model configurations of DUALAD. As in the main paper, we indicate all stage-I models that are trained on perception tasks only e.g. object detection, map segmentation and multiple object tracking as DUALAD-I and the configuration that was trained on all tasks in an end-to-end fashion as DUALAD-II respectively. Furthermore, we adopt the notation introduced in Table 6 to denote different configurations of DUALAD. The version marked with $\emptyset$ does not use the proposed dynamic-static cross-attention, while $\updownarrow$ describes a version that uses bidirectional stream interaction by using global attention for the interaction from the static to the dynamic stream. The version of our model that is trained on the reduced sensor set by using front and back facing cameras in an alternating fashion only is indicated with $\ominus$.

Object Detection: A detailed evaluation of all metrics specified in the official nuScenes detection benchmark [21] is shown in Table 8. For detailed metric definitions, we kindly refer to [1, 21].

Map Segmentation: The results for all model configurations on map segmentation are shown in table Table 9. The evaluation is performed for four different classes as proposed in UniAD [10] and we compute the IoU between predicted and ground truth segmentation maps.

Multiple Object Tracking: A detailed evaluation of all metrics specified in the official nuScenes tracking benchmark [22] is shown in Table 10. For detailed metric definitions, we kindly refer to [1, 22].

Motion Prediction: A detailed evaluation of motion prediction results for all dynamic classes of the nuScenes dataset [1] is shown in Table 11. As in UniAD [10] we adopt a confidence threshold $c_{\text{motion}} = 0.4$ during inference to select object queries that are passed to the motion head.

7.1. Discussion of design choices

The extensive ablations on various tasks and configurations of our proposed approach (see Table 8, Table 9, Table 10) validate our different design choices. Our model consistently benefits from temporal information and the

Table 8. Object Detection Results.

Name	Temporal BEV	Sensor Drop	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	NDS↑
DUALAD-I	✗	✗	46.93	0.62	0.27	0.39	0.27	**0.18**	56.16
DUALAD-I∅	✓	✗	47.74	0.62	0.27	0.45	0.28	0.19	55.78
DUALAD-I↕	✓	✗	49.37	0.58	0.27	0.39	0.26	0.20	57.65
DUALAD-I¶	✓	✗	48.21	0.60	0.27	**0.32**	0.28	0.20	57.44
DUALAD⊖	✓	✓	42.86	0.65	0.28	0.47	0.32	0.19	52.22
DUALAD-I	✓	✗	**49.56**	0.58	**0.26**	0.40	**0.26**	0.20	**57.81**
DUALAD-II	✓	✗	48.16	**0.57**	0.27	0.41	0.29	0.19	56.68

Table 9. Map Segmentation Results.

Name	Temporal BEV	Sensor Drop	Lanes↑	Drivable↑	Divider↑	Crossing↑
DUALAD-I	✗	✗	31.73	67.52	26.57	10.99
DUALAD-I∅	✓	✗	33.97	69.35	29.49	12.33
DUALAD-I↕	✓	✗	33.86	67.78	29.11	12.18
DUALAD-I¶	✓	✗	34.26	69.71	29.71	**13.87**
DUALAD⊖	✓	✓	31.53	66.60	26.77	10.14
DUALAD-I	✓	✗	**34.68**	**70.50**	**30.29**	12.82
DUALAD-II	✓	✗	34.17	70.01	29.96	12.25

proposed dynamic-static cross-attention. Adding another cross-attention block to perform bidirectional interaction does not significantly improve the performance of static map perception or overall temporal consistency, which is in line with our hypothesis that map segmentation might not benefit from dynamic agent perception. We leave the investigation of other interaction designs and other dense tasks that depend on the dynamic agent perception e.g. free-space estimation for future work.

The stage-II configuration of our approach yields a slightly decreased perception performance when compared to the stage-I model. This could result from the fact that in stage-II the model might focus on certain scene parts that are more relevant for the currently planned trajectory. Additionally, a fast detection of highly dynamic agents and temporal consistency might be crucial for longer planning horizons, which is in line with the improvements of the stage-II model in terms of *Track Initialization Duration* (TID) and *Longest Gap Duration* (LGD) as shown in Table 10.

The DUALAD-I⊖ version of our model that only has access to front or back facing cameras in an alternating fashion maintains high temporal consistency by query propagation even without sensor data for some areas in the scene. We refer to the attached video for a qualitative example. However, the initial detection of newly appeared object is not possible if no sensor data for the corresponding scene area is available or consistent tracking might be challenging, especially for highly dynamic or hardly visible agents in the scene. Since our base model especially improves over previous approaches in such challenging cases, this explains the drop in perception performance by -6.7 mAP and -10.7 AMOTA respectively (see Table 8, Table 10).

7.2. Runtime Analysis

We evaluate the runtime of the stage-II configuration of DUALAD. The results of the entire system as well as the runtime of the intermediate task modules are shown in Table 12. DUALAD runs with 4.12 FPS on a single NVIDIA A100 GPU. The dual stream transformer uses a significant amount of the model's total runtime due to the expensive attention operations from object queries and BEV-queries to sensor data. Since all downstream tasks use the resulting representations, the task heads only add a small amount of additional runtime. Please note that our codebase contains various operations which could be further optimized. However, improving the runtime and memory requirements of end-to-end approaches remains a challenging topic for large scale application of such approaches.

7.3. Integration to VAD [11]

The version of our model that is based on VAD [11] is denoted as DUALVAD, please note that we report the performance of the stage-II model to allow for a fair comparison with the provided model in [28]. In contrast to the other

Table 10. Multiple Object Tracking Results.

Name	Temporal BEV	Sensor Drop	AMOTA↑	AMOTP↓	RECALL↑	MT↑	ML↓	FAF↓	IDS↓	FRAG↓	TID↓	LGD↓
DUALAD-I	✗	✗	51.63	1.16	59.69	3006	2104	49.08	658	671	1.25	1.96
DUALAD-I∅	✓	✗	51.94	1.13	59.27	3107	2148	48.37	769	657	1.24	1.84
DUALAD-I↕	✓	✗	54.39	1.09	**61.11**	3232	2077	46.76	**588**	**580**	1.14	1.70
DUALAD-I¶	✓	✗	52.32	1.13	60.74	3272	**1908**	49.46	726	695	1.09	1.67
DUALAD⊖	✓	✓	44.39	1.22	53.96	2658	2476	53.57	940	936	1.44	2.01
DUALAD-I	✓	✗	**55.09**	**1.09**	60.71	**3279**	2031	**46.21**	663	588	1.12	1.70
DUALAD-II	✓	✗	52.57	1.11	59.62	3159	2166	46.25	774	593	**1.07**	**1.61**

Table 11. Motion prediction results of DUALAD-II for all object categories on the nuScenes benchmark [22].

Name	EPA↑	minADE↓	minFDE↓	miss rate↓
Car	54.97	0.35	0.39	0.035
Truck	43.12	0.37	0.38	0.017
Bus	42.31	0.51	0.56	0.057
Trailer	26.79	0.55	0.53	0.017
Pedestrian	45.28	0.46	0.61	0.003
Motorcycle	39.02	0.32	0.37	0.011
Bicycle	36.89	0.28	0.30	0.002

Table 12. Runtime evaluation of DUALAD-II on a single NVIDIA-A100 for 500 frames of the nuScenes validation set. Misc describes various non-optimized computations e.g. bounding box decoding and positional encodings.

Module	Runtime (ms) ↓
Image Backbone	19
Dual Stream Transformer	59
Detection Head	17
Map Head	23
Motion Head	24
Planning Head	39
Misc	80
Total	242

configurations, VAD relies on a ResNet-50 [9] as image backbone, an input resolution of 1280×720 pixels [11, 28] and a shorter detection range around the ego vehicle of $[-30\,\mathrm{m}, 30\,\mathrm{m}]$ in x and $[-15\,\mathrm{m}, 15\,\mathrm{m}]$ in y respectively. A detailed evaluation of the perception performance is given in Table 14. DUALVAD outperforms VAD [11] by $+2.7$ mAP for dynamic object perception and achieves a slightly higher vectorized map perception performance while also heavily improving downstream tasks such as motion prediction (see Table 4) and open-loop planning (see Table 5). The runtime of DUALVAD-II is shown in Table 13. In this configuration, our model runs at $3.32\,\mathrm{FPS}$ on a single NVIDIA A100 GPU. Due to the larger input image size, the runtime of the dual stream transformer increases significantly as compared to our base configuration.

Table 13. Runtime evaluation of DUALVAD-II on a single NVIDIA-A100 for 500 frames of the nuScenes validation set. Misc describes various non-optimized computations e.g. bounding box decoding and positional encodings.

Module	Runtime (ms) ↓
Image Backbone	11
Dual Stream Transformer	114
Detection Head	58
Map Head	4
Motion Head	7
Planning Head	3
Misc	104
Total	301

7.4. Qualitative Results

Together with this document, we provide a video that shows qualitative results of our approach for various scenes from the nuScenes validation set. Those include complex traffic scenes, a setting with unsynchronized sensors, challenging lighting and adverse weather conditions and results for the vectorized map representation. DUALAD-II demonstrates robust and consistent performance for all perception tasks, as well as downstream performance for motion prediction and open-loop planning.

Table 14. Perception Results for VAD [11] based models. *Results taken from official repository. mAP_{Map} denotes the mAP of vectorized map perception as defined in [11, 16].

Name	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	NDS↑	mAP_{Map}↑
VAD [11]*	33.92	0.59	0.28	**0.53**	0.40	**0.23**	46.02	47.5
DUALVAD-II	**36.64**	**0.59**	**0.27**	0.57	**0.35**	0.23	**48.00**	**47.9**