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

Simon Doll[1,2], Niklas Hanselmann[1,2], Lukas Schneider[1], Richard Schulz[1], Marius Cordts[1], Markus Enzweiler[3], Hendrik P.A. Lensch[2]

[1]Mercedes-Benz AG, [2]University of Tübingen, [3] Esslingen University of Applied Sciences

Abstract

State-of-the-art approaches for autonomous driving integrate multiple sub-tasks of the overall driving task into a single pipeline that can be trained in an end-to-end fashion by passing latent representations between the different modules. In contrast to previous approaches that rely on a unified grid to represent the belief state of the scene, we propose dedicated representations to disentangle dynamic agents and static scene elements. This allows us to explicitly compensate for the effect of both ego and object motion between consecutive time steps and to flexibly propagate the belief state through time. Furthermore, dynamic objects can not only attend to the input camera images, but also directly benefit from the inferred static scene structure via a novel dynamic-static cross-attention. Extensive experiments on the challenging nuScenes benchmark demonstrate the benefits of the proposed dual-stream design, especially for modelling highly dynamic agents in the scene, and highlight the improved temporal consistency of our approach. Our method titled DualAD not only outperforms independently trained single-task networks, but also improves over previous state-of-the-art end-to-end models by a large margin on all tasks along the functional chain of driving.

1. Introduction

Autonomous systems have evolved from strictly modular and largely hand-crafted pipelines towards a more holistic learning-centric paradigm [3, 25]. While the former relies on explicitly defined interfaces between modules, the latter tackles the entire driving task in an end-to-end fashion. Nevertheless, recent work has shown the benefits of retaining a modular structure including typical sub-tasks such as perception, prediction and planning while allowing latent features to serve as interfaces between the modules [10, 11].

In contrast to independent, task-specific modules with fixed pre-defined interfaces, an end-to-end approach enables the joint optimization of the entire pipeline, learning not only the parameters in each module but also *the interfaces between them*. The chosen space of each module's latent representation restricts the set of interfaces which can be learned, allowing to model inductive biases about the scene structure, e.g. consistent motion of dynamic elements, or to incorporate task specific properties. However, this places additional importance on choosing well-suited intermediate representations, since they heavily affect information flow and the performance of subsequent modules. Hence, these representations should be carefully tailored to their corresponding semantic entities in the driving scene to achieve a high performing end-to-end architecture.

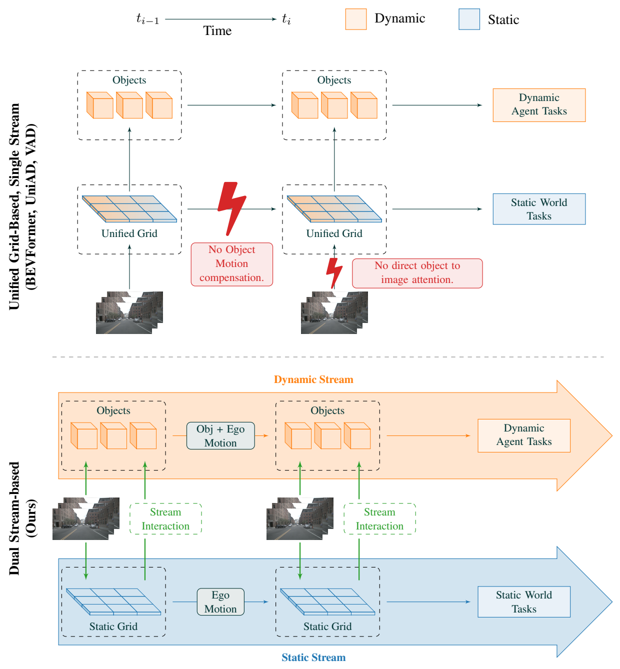


Figure 1. **Comparison of Representation Design** of unified grid-based approaches and our dual-stream design. By explicitly disentangling dynamic and static representations, the dynamic stream can aggregate highly descriptive features. This is achieved through direct attention to image features, as well as explicit compensation for object and ego motion, which is not feasible with unified grids.

To model dynamic agents in the scene, a prevalent approach is to leverage attention with object-centric queries that detect an individual object in the environment [6, 18, 32]. Furthermore, recent works [7, 30, 36] have demonstrated the benefits of incorporating temporal information to consistently model object dynamics and to account for temporal occlusions. In such work, object-queries provide dedicated latent representations that each describe a single object. Its belief state can then be propagated through time by explicitly compensating for the ego and estimated object motion between two consecutive timestamps [7, 30].

The most common alternative is to use bird's-eye view (BEV)-grid queries [10, 11, 14] as an intermediate representation, with subsequent tasks relying solely on this representation. However, such a grid is not coupled to semantic instances and instead represents a spatial area of the scene. Hence, the motion of agents cannot be explicitly modeled and compensated for, see Fig. 1. This is due to the fact that each grid-cell could potentially represent multiple entities with different rigid motion transforms or even completely static elements, depending on grid resolution and object sizes. While grid-based representations are well-suited for static world perception [14, 16], exclusively relying on them to aggregate sensor measurements and temporal information hampers the perception of highly dynamic agents.

Contributions: In this work, we propose a dual-stream approach to leverage the potential of object-centric representations for dynamic agents combined with a BEV-grid representation for static scene elements. This dual-stream design explicitly applies object and ego motion compensation to dynamic agents and allows object-queries and BEV-queries to simultaneously attend to the camera images of the current timestamp. Besides self-attention and cross-attention with camera images, we introduce a new dynamic-static cross-attention-block that allows object-queries to attend to the BEV-queries fostering the consistency between the streams.

Our proposed approach termed DUALAD allows for robust and temporally consistent perception. On the challenging nuScenes dataset [1] DUALAD outperforms specialized *state-of-the-art* (SOTA) models for various perception tasks by a large margin. The integration with recent end-to-end frameworks [10, 11] reveals the importance of disentangled representations for dynamic agents and static world elements, and exhibits significant performance gains along the entire functional chain. Extensive ablation studies highlight the importance of the dual-stream design for all driving tasks, while especially improving temporal consistency and the perception of highly dynamic agents.

2. Related Work

Accurate and consistent perception forms the basis for autonomous driving. We structure related literature into three categories: (i) specialized models for *dynamic agents* that perform 3D object detection and 3D multiple object tracking, (ii) models that reason about *static scene elements* and perform online mapping, and (iii) *multi-task end-to-end* models that jointly perform the aforementioned tasks in a single model and can be optimized end-to-end.

Perception of Dynamic Agents: Based on pioneering works [2, 32], recent specialized models for 3D object detection utilize a transformer-based architecture with a set of object-queries to detect objects in the scene [6, 14, 18, 33]. Several extensions have been proposed e.g. to reduce the memory footprint and to increase the convergence speed, resulting in improved overall performance [13, 38]. Incorporating temporal information for the perception of dynamic agents can be achieved by query propagation and therefore implicit tracking [14, 24, 30] combined with various tracking-by-detection approaches [29, 34], or by tracking-by-attention [7, 36]. For such query propagation, it is crucial to follow an object-centric paradigm. This allows to aggregate descriptive features for each object by performing attention directly between object-queries and sensor measurements, as well as explicitly compensate for the motion of objects between consecutive time steps [7, 30].

Another line of work utilizes an intermediate grid of BEV-queries to propagate information through time [10, 11, 14]. In such approaches, each BEV-query always represents the same area in the grid and is not coupled to a specific semantic element. Dynamic agents are then detected using queries attending to this grid. However, since compensating for the motion of dynamic agents in the grid is not directly possible, we opt for an object-centric approach to model dynamic agents in our dual-stream design.

Perception of Static Scene Elements: Inspired by recent works on 2D panoptic segmentation [15], current works that perform online map segmentation rely on BEV-grid-queries, coupled with a transformer-decoder architecture to perform BEV map segmentation [10, 14].

Another class of approaches tries to model map perception tasks in a vectorized fashion, where map elements are directly modeled as a sequence of points, e.g. by leveraging map queries [16, 19, 28]. As both variants rely on a temporal BEV-grid to achieve a temporally consistent performance, we follow this concept for static world perception.

Multi-Task End-to-End Models: Most recently, different approaches [10, 11] proposed to model the driving task as a modular pipeline that is trainable end-to-end. This allows to optimize the individual modules as well as their interfaces towards the final driving task. The modules are typically connected by transformer mechanisms, effectively defining interfaces in terms of query, key and value triplets.

Inspired by the aforementioned works, we propose a dual-stream transformer that can be used as the foundation for various perception tasks as well as for end-to-end multi-task driving. We simultaneously use object-centric queries to represent dynamic agents in the scene, while modelling static scene elements with BEV-grid-queries. This explicitly disentangles the representation of static and dynamic elements in the scene, resulting in a higher temporal consistency, especially for highly dynamic agents. The resulting architecture combines the potential of SOTA approaches for dynamic object perception as well as static perception in a single model, and can directly be integrated with recent multi-task models to train the entire stack end-to-end.

3. Method

As shown in Fig. 2a, our proposed approach DUALAD comprises a transformer-decoder-based perception architecture that uses two streams to explicitly model dynamic objects in an object-centric and static scene elements in a grid-based fashion. The resulting dynamic and static world representations enable various tasks relevant to driving such as 3D object detection and tracking, map segmentation, motion prediction as well as planning. Furthermore, our approach permits an end-to-end optimization of the entire driving stack as proposed in [10, 11].

At each time step t a set of N multi-view camera images $\mathcal{I}_t$ is fed into a shared image feature extractor. The resulting image features $\mathcal{F}_t$ are used by both, the dynamic object as well as the static stream. The former reasons about dynamic agents in the scene, like cars or pedestrians. These agents are represented by a set of object-queries $q_{\text{obj}} \in \mathcal{Q}_{\text{obj}}$ that can be decoded into a bounding box $b_t = \left[x, y, z, w, l, h, \theta, v_x, v_y\right]$ as well as the predicted class c of the agent. In parallel, a grid of BEV-queries $q_{\text{BEV}} \in \mathcal{Q}_{\text{BEV}}$ with dimensions $H_{\text{BEV}} \times W_{\text{BEV}}$ uses $\mathcal{F}_t$ to reason about the static scene. The resulting BEV representation is used to perform panoptic segmentation of the road topology, e.g. drivable space or lane markings, utilizing a segmentation head as proposed in [10, 15].

Interaction between the two streams is enabled by novel dynamic-static cross-attention blocks (see Fig. 2b) where the object-queries q_{obj} attend to the BEV-queries q_{BEV} representing the static scene structure. Since the temporal representations for the dynamic and the static world are disentangled, we can explicitly compensate object and ego motion for the object-centric queries of dynamic agents while the static BEV-queries only require a propagation that is dependent on the motion of the ego vehicle.

3.1. Dual Stream Design for End-to-End Driving

Finding a well-suited representation for the belief state of the scene is key for any transformer-based end-to-end trainable driving stack as motivated in Section 1. In comparison to traditional pipelines, the end-to-end paradigm allows the interfaces to be optimized towards subsequent modules in the pipeline. Nevertheless, the chosen space of latent representations heavily affects the ability to model the relevant semantic entities and their relations [11, 30].

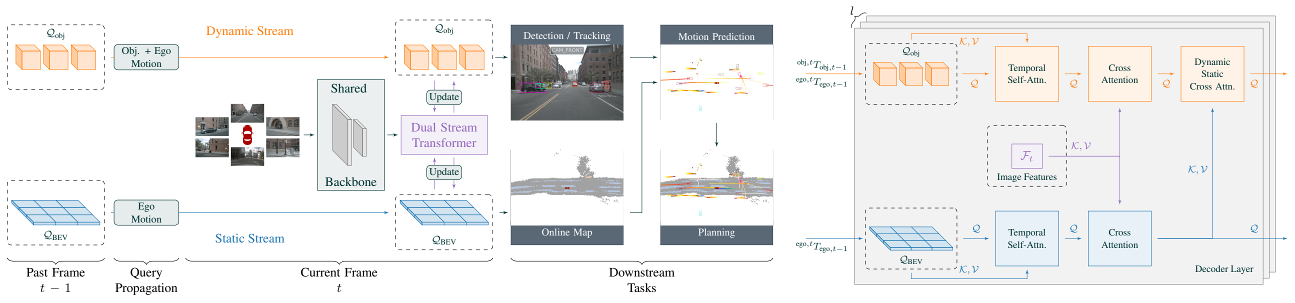
Whereas unified BEV-grid representations can appropriately handle static content, representing highly dynamic objects in a BEV-grid is ill-posed, since each cell might describe multiple entities with different motion patterns, static scene elements or even a combination of both. We therefore argue that dynamic objects and static scene content should be represented separately, and propose a dual-stream architecture consisting of a dynamic and a static stream.

Dynamic Stream: In DUALAD dynamic objects are modelled with an object-centric representation by using a single object-query q_{obj} to describe an individual object in the scene [6, 7, 18, 30, 32]. To obteain a highly descriptive representation, we propose that each object-query should directly perform cross-attention to the image features $\mathcal{F}_t$. In contrast to unified methods in which only the BEV-grid queries directly attend to images [10, 11, 14], this enables to exploit the high spatial resolution of the image features for more precise detection and tracking.

Following the arguments in [7, 30], we propose to propagate the latent queries q_{obj} to the next timestamp by compensating for the motion between the two timestamps via a latent transformation that depends on the geometric motion. In contrast to static scene parts, the observed motion of objects consists of two separate components: (1) the motion ${}^{\text{ego}_{t+1}}\mathbf{T}_{\text{ego}_t}$ of the ego vehicle and (2) the motion ${}^{obj_{t+1}}\mathbf{T}_{obj_t}$ of the dynamic object itself. For more details on query propagation, we kindly refer the reader to [7, 30].

In detail, our approach uses the top-k propagated object-queries of each time step as priors in the subsequent frame, following an implicit tracking approach as in Stream-PETR [30] to account for temporary occlusions and to consistently track objects in the scene. In contrast to tracking-by-attention [7, 10, 36] in which only matched objects are propagated to the next frame, this allows our model to maintain multiple hypotheses for the same object and does not require explicit track handling. To obtain explicit object identities, our model can be combined with any tracking-by-detection approach.

Static Stream: We use a BEV grid-based representation to model static scene elements. A dense, spatially regular representation is well-suited for non-moving objects in the surrounding area. Since all elements in the grid are assumed to be static, updates over time are incorporated by applying a rigid transform to the BEV-grid computed from the ego motion ${}^{\text{ego}_{t+1}}\mathbf{T}_{\text{ego}_t}$. We sample grid features differentiably via interpolation and use deformable temporal grid-attention as proposed in [14]. Map segmentation is then performed with a decoder-only segmentation head [15, 16]. This significantly simplifies the map segmentation head as compared



(a) Pipeline overview

(b) Dual-stream transformer

Figure 2. **DUALAD Architecture**: Two separate representations are chosen for dynamic agents ($\mathcal{Q}_{\text{obj}}$) and static elements ($\mathcal{Q}_{\text{BEV}}$) as shown in Fig. 2a. Self- and cross-attention is simultaneously performed in the proposed dual-stream transformer as shown in Fig. 2b, paired with the novel dynamic-static cross-attention block to allow the dynamic agents to benefit from the inferred scene structure.

to unified grid-based approaches, since those typically require an additional grid-encoder [10, 14].

3.2. Modelling Interactions between the Dynamic and Static World

Explicitly disentangling dynamic agents and static scene elements results in two independent streams of the model. Both streams rely on shared image features $\mathcal{F}_t$, but perform self-attention and cross-attention to these features separately. To enable the network to leverage mutual information between static scene elements and dynamic agents, we propose an additional attention block that performs dynamic-static cross-attention between the streams.

As shown in Fig. 2b, this is achieved by performing deformable attention [38] between object-queries q_{obj} and BEV-queries q_{BEV} of the current timestamp that are close to the position of the object [14]. In doing so, the dynamic objects can infer their state update more precisely by considering not only the sensor information but also the aggregated static BEV-grid, e.g. by incorporating the estimated information on road layout and lane topology.

Movable Belief State Through Space and Time: Our dual stream design enables incorporating even unsynchronized sensor input. Whenever sensor information is available, potentially at arbitrary time intervals, the belief state of static and dynamic parts can be propagated to that timestamp considering ego and object motion. The novel sensor data is then easily integrated via cross-attention to the available image features to update the inferred scene state.

Hence, our approach facilitates incorporating sensor measurements at different points in time while simultaneously keeping a temporally consistent representation of the scene. Our model can also handle cases, where the set of sensors varies at each time step. This is especially relevant for non-synchronized sensors, e.g. due to different sensing rates, or even sensor failures. Additionally, this enables to use ground truth annotations that are synchronized with only a subset of the sensors, as well as getting a model output at timestamps between sensor measurements, which might be beneficial for real-time applications [31].

4. Experiments

We evaluate the performance of DUALAD on the challenging and well-established nuScenes dataset [1]. Additionally, we integrate our proposed approach into two SOTA end-to-end trainable driving frameworks, i.e. UniAD [10] and VAD [11]. We perform extensive ablation studies to evaluate the effect of our design choices and provide additional insights as well as qualitative results.

Dataset: We utilize the large-scale nuScenes dataset [1] consisting of 1000 scenes and use the official train- and val-set split. We adopt the official task definitions for the object detection task [21] and object tracking task [22], respectively, and follow other recent works [8, 10, 11] for the definition of the motion prediction and planning objectives.

Metrics: For object detection, we report the main metrics *mean Average Precision* (mAP) and *nuScenes Detection Score* (NDS) computed on all ten classes of the dataset, as well as true positive metrics such as the *mean Average Translation Error* (mATE), *mean Average Orientation Error* (mAOE), and *mean Average Velocity Error* (mAVE) as defined in [21]. For object tracking, we follow the official metric definition in [22] and report *Average Multi Object Tracking Accuracy* (AMOTA) and *Average Multi Object Tracking Precision* (AMOTP) as well as recall and *number of identity switches* (IDS). For map segmentation, we closely follow [10] and report BEV segmentation *Intersection over Union* (IoU) for different classes. We refer the reader to [10] for additional details on the metric and class definitions. For motion prediction, we report the *End-to-end Prediction Accuracy* (EPA) [8] as main metric as well as the true positive metrics *minimum Average Displacement Error* (minADE) and *minimum Final Displacement Error* (minFDE). For open-loop planning, we report the L2 distance to the ego trajectory as well as collision rates for 1 s and 3 s, respectively. A more detailed evaluation including additional metrics for the different configurations of our approach can be found in the supplementary.

Training Configuration: We closely follow the settings in [10, 11, 30] to increase comparability. Unless otherwise specified, we utilize a VovNet-V2-99 [12] and an image resolution of 800×320 pixels. For details on the backbone and FPN [17] configuration, we refer the reader to [26, 30]. Following [30], we utilize streaming video training. All models are trained for 24 epochs utilizing a batch size of eight on eight NVIDIA A100 GPUs with AdamW [20], a learning rate of $2e^{-4}$ and a cosine annealing schedule. DUALAD performs object detection, tracking, map segmentation, and motion prediction, as well as open-loop planning. Note that all tasks are performed jointly in one multi-task model. Following [10], we train the model in a two-stage fashion: the stage-I model only performs object detection, tracking and map segmentation, while the stage-II model is optimized for all tasks in an end-to-end fashion with a frozen image backbone for numerical stability. In all perception experiments, we append -I or -II for clarity e.g. DUALAD-II.

Baselines: Given that our approach follows the object-query propagation technique used by StreamPETR [30], and considering that StreamPETR reaches SOTA performance on the nuScenes benchmark [21], we choose it as the main baseline for dynamic object perception. To demonstrate the performance gains of our proposed architecture along the entire functional chain, we evaluate the performance of DUALAD on downstream tasks for driving, such as motion prediction and open-loop planning. We choose the two recent SOTA approaches UniAD [10] and VAD [11] as baselines since they perform all tasks in an end-to-end trainable fashion while also following the two-stage training paradigm. If not all metrics are reported in the corresponding publications, we utilize the published training logs and code to reproduce the results. For tasks without an official benchmark, we adopt the evaluation scheme of [10, 11]. It has to be noted that VAD [11] utilizes a ResNet-50 and a smaller detection range for the perception and motion prediction tasks. For all comparisons with VAD [11], we configure our model to exactly follow their settings for a fair comparison. We refer the reader to [11, 28] for additional details on the configuration and used detection ranges.

4.1. Perception Sub-Task Results

In this section, we analyze the performance of our stage-I model trained on perception tasks only, to allow for a fair comparison with SOTA models specialized for perception.

Object Detection: The object detection performance on the nuScenes validation set [1] is shown in Table 1. Compared to other multi-task models like UniAD [10], DUALAD yields an improvement of +10 (+20 %) in terms of mAP and +7.2 (+12 %) in NDS respectively. Compared to StreamPETR [30], which is only trained on object detection, DUALAD gains an improvement of +1.3 mAP and +0.7 NDS, resulting in SOTA performance for object detection. While our model builds on StreamPETR, we attribute the key improvements over StreamPETR to the newly introduced dynamic-static cross-attention that allows object-queries to benefit from the static scene structure.

Table 1. **Object Detection**. DUALAD outperforms task-specific models as well as end-to-end models on all metrics. We report performance for perception as well as stage-II results for end-to-end models. *Results taken from official repository [27]. ¶ indicates a version that uses a ResNet-101 [9], as in UniAD [10].

Name	mAP↑	mATE↓	mAOE↓	mAVE↓	NDS↑
BEVFormer[14]	41.6	0.67	0.37	0.27	51.7
SteamPETR[30]	48.2	0.60	0.37	0.26	57.1
UniAD-I*[10]	39.5	0.66	0.36	0.40	50.6
DUALAD-I¶	48.2	0.59	**0.32**	0.28	57.4
DUALAD-I	**49.5**	**0.57**	0.39	**0.26**	**57.8**
UniAD-II[10]	38.1	0.68	0.38	0.38	49.8
DUALAD-II	48.1	0.57	0.41	0.28	56.6

Map Segmentation: The results for map segmentation of different classes are shown in Table 2. DUALAD yields comparable performance to SOTA approaches on all classes while improving the lane segmentation by +2.9 IoU as compared to UniAD using the same image backbone. As indicated in [10], the observed improvements in lane segmentation are vital for the accurate perception of dynamic agents. We want to highlight that our two-stream design allows the use of a single static BEV encoder instead of having a unified BEV encoder and an additional static map encoder. This results in 2.7 M (−15 %) fewer parameters for map segmentation as compared to UniAD.

Multiple Object Tracking: The tracking performance of our model is shown in Table 3. In contrast to explicit tracking as in [7, 10] our model only performs implicit tracking through query propagation. We utilize the widely adopted tracking approach presented in [34] for a fair comparison to StreamPETR [30]. DUALAD reaches SOTA performance in terms of AMOTA and outperforms specialized tracking approaches like PF-Track [23] by a large margin. Compared to UniAD [10], our approach heavily improves the AMOTA by +15.8 (+28 %) and by +2.5 (+4 %) compared to the specialized StreamPETR, respectively. In particular, the dual-stream layout leads to a higher temporal consistency of tracked objects, manifesting in a reduction of IDS by 25 % compared to UniAD [10] and StreamPETR [30].

Table 2. **Map Segmentation**. DUALAD achieves competitive performance, especially improving the segmentation of lanes. ¶ indicates a version that uses a ResNet-101 [9] as in UniAD [10], $^-$ a version only trained on the mapping task without dynamic agents. We report performance for perception as well as stage-II results for end-to-end models. *Results taken from official repository [27]. Segmentation IoU(%) is reported for different classes.

Name	Lanes↑	Drivable↑	Divider↑	Crossing↑
BEVFormer[14]	23.9	**77.5**	-	-
BEVerse[37]	-	-	30.6	**17.2**
UniAD-I* [10]	31.3	69.1	25.7	13.8
DUALAD-I¶	34.2	69.7	29.7	13.8
DUALAD-I	34.6	70.5	30.2	12.8
DUALAD-I¶$^-$	**35.6**	71.1	**32.3**	15.1
UniAD-II*[10]	31.2	69.1	25.9	14.3
DUALAD-II	34.1	70.0	29.9	12.2

Table 3. **Multiple Object Tracking**. DUALAD achieves high temporal consistency, heavily outperforming previous SOTA approaches. ¶ indicates a version that uses a ResNet-101 [9], as in UniAD [10], $^+$ tracking-by-detection approach reimplemented with BEVFormer [14] in [10]. We report performance for perception as well as stage-II results for end-to-end models. *Results taken from official repository [27].

Name	AMOTA↑	AMOTP↓	Recall↑	IDS↓
ImmortalTrack$^+$[29]	37.8	1.11	47.8	936
PF-Track[23]	47.9	1.22	59.0	**181**
StreamPETR[30]	52.6	1.12	59.9	886
UniAD-I*[10]	39.3	1.29	48.1	894
DUALAD-I¶	52.3	1.12	60.7	726
DUALAD-I	**55.1**	**1.08**	**60.7**	663
UniAD-II*[10]	36.3	1.34	45.9	1177
DUALAD-II	52.6	1.11	59.6	774

Runtime: DUALAD-I perception takes an average runtime of 193 ms. The new Dynamic-Static Cross-Attention only adds 2.12 ms per block, corresponding to 6 % of the total runtime (see Supplementary). A small configuration with 107 ms latency reaches 40.1 % mAP and therefore still outperforms UniAD [10] while being four times faster.

4.2. Integration to End-to-End Pipelines

To demonstrate the performance gains for downstream tasks like motion prediction and planning, we integrate our proposed dual-stream architecture into recent end-to-end trainable driving frameworks that reach SOTA results: UniAD [10] and VAD [11]. Due to the training focus on the final motion prediction and planning performance, we observe, similar to UniAD [10], a slight degradation in perception performance when training in the end-to-end setting as shown in Table 1, Table 2 and Table 3.

Table 4. **Motion Prediction**. DUALAD remarkably improves the motion prediction task within different frameworks. V denotes the vehicle category and P pedestrians, respectively. *Results taken from official repository [27, 28]. Please note that VAD [11] and hence our integration DUALVAD use a smaller detection range and that results are not directly comparable with other approaches.

	EPA↑		minADE↓		minFDE↓	
Name	V	P	V	P	V	P
VIP3D[8]	22.2	-	2.05	-	1.95	-
UniAD*[10]	45.6	35.5	0.71	0.78	1.02	1.05
DUALAD	52.4	45.2	0.68	**0.63**	1.08	0.89
VAD* [11]	64.7	47.4	0.68	0.67	0.92	**0.84**
DUALVAD	**69.8**	**47.9**	**0.60**	0.67	**0.83**	0.85

Table 5. **Open-Loop Planning**. DUALAD achieves lowest L2 error and collision rate, especially for longer planning horizons. Avg describes the mean over the 1s, 2s and 3s values, DUALVAD a version of our model that follows the VAD [11] framework.

	L2 (m) ↓			Col (%) ↓		
Name	1s	3s	Avg	1s	3s	Avg
UniAD[10]	0.48	1.65	1.03	0.05	0.71	0.31
DUALAD	0.56	1.55	1.03	**0.03**	**0.35**	**0.17**
VAD[11]	0.41	1.05	0.72	0.07	0.41	0.22
DUALVAD	**0.30**	**0.82**	**0.55**	0.11	0.36	0.22

Motion Prediction: The results for motion prediction are shown in Table 4. Our model significantly outperforms UniAD by +6.8 (+12 %) EPA for the vehicle class and by +9.7 (+21 %) for pedestrians, respectively. A similar effect is observed for the vectorized framework VAD [11] where our model improves motion prediction by +5.1 (+7 %) EPA for vehicles. We attribute this to improved modelling of dynamic agents in the scene and improved motion queues by direct object to image attention.

Open-Loop Planning: The evaluation of the open-loop planning performance is provided in Table 5. While we report those results for completeness, we want to highlight the issues on open-loop planning in nuScenes [1] recently identified in [35]. We integrate our approach into the planning modules proposed in [10] and [11] that do not use the ego status as direct input to planning and follow their corresponding evaluation protocols [27, 28]. As shown in Table 5, DUALAD reaches comparable performance in terms of $L2$ distance compared to UniAD [10] and heavily reduces the collision rate up to a factor of two for longer planning horizons. Similarly, for VAD [11] the $L2$ error is reduced by up to 0.23 m (−21 %) depending on the planning horizon. This is in line with our model design since improv-

Table 6. **Ablation on the Interaction Design**. † denotes UniAD with the detection head of StreamPETR [30] instead of the tracking-by-attention head proposed in [36]. ∅ denotes a variant of our approach without stream interaction and ↕ a variant that uses bidirectional stream interaciton.

Name	w/Det	w/Map	Temporal BEV	Obj2Img Attn	#Params	mAP↑	NDS↑	Lanes↑	AMOTA↑	IDS↓
UniAD-I [10]	✓	✓	✓	✗	-	39.5	50.6	29.3	39.3	894
StreamPETR [30]	✓	✗	✗	✓	-	48.2	57.1	-	52.6	886
UniAD-I†[10]	✓	✓	✓	✗	111M	43.3	51.2	33.9	45.0	1764
UniAD-I†[10]	✓	✓	✗	✗	111M	45.3	54.9	34.3	49.4	779
DUALAD-I	✓	✓	✗	✓	118M	46.9	56.1	31.7	51.6	658
DUALAD-I∅	✓	✓	✓	✓	113M	47.7	55.7	33.9	51.9	769
DUALAD-I↕	✓	✓	✓	✓	120M	49.3	57.6	33.8	54.4	**588**
DUALAD-I	✓	✓	✓	✓	118M	**49.5**	**57.7**	**34.6**	**55.1**	663

ing temporal consistency, especially for dynamic agents, might become more relevant for longer planning horizons.

Qualitative Results: Fig. 3 visualizes the performance of DUALAD in a complex traffic scene. The proposed dual-stream design enables a temporally consistent perception of the surrounding area, including highly dynamic agents and allows for precise motion prediction and planning. A comparison to the perception of UniAD [10] with respect to highly dynamic agents is shown in Fig. 4. Additional examples for both integrated frameworks [10, 11] can be found in the supplementary.

4.3. Ablations

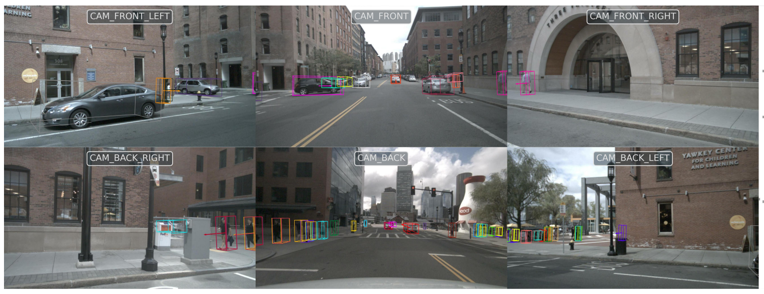
Effect of Interaction Design: An evaluation of the different design choices of DUALAD is shown in Table 6. First, using a version of our approach with two separate streams without the proposed dynamic-static cross-attention module especially decreases the object detection and tracking performance. This observation confirms the benefits of the interaction with the static branch for dynamic agents. Second, using a bidirectional stream interaction, where also the static BEV-queries attend to the dynamic agents, does not yield significant improvements. This is in line with our hypothesis that it is sufficient for the static world representation to perform cross-attention to the images paired with temporal self-attention. Third, we incorporate StreamPETR's query propagation [30] as used in our approach to UniAD [10], which yields consistent improvements but heavily increases the IDS which might be a result of the simple greedy tracker [34]. Fourth, DUALAD benefits as expected from using temporal attention for the BEV-queries, but we observe the opposite for UniAD [10]. A version without temporal attention within the unified BEV-grid in UniAD† has a performance for dynamic agent perception that is increased by $+3.7$ NDS and 4.4 AMOTA, respectively, while IDS are drastically reduced by $55\,\%$. This supports our claim that the unified grid is not well-suited to propagate information about dynamic agents through time.

Table 7. **Ablation on Temporal Consistency**. DUALAD achieves consistent tracks even without sensor measurements from each sensor in each frame. We mimic the effect of non-synchronized sensors by using the front and back facing cameras in an alternating fashion (denoted by ⊖).

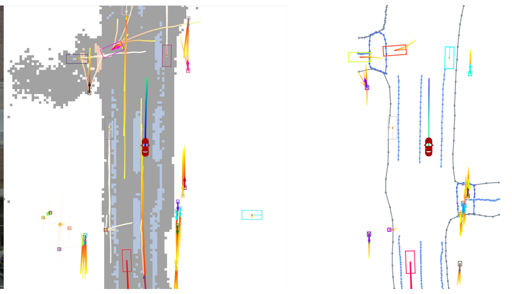
Name	mAP↑	Lanes↑	AMOTA↑	IDS↓
UniAD-I[10]	39.5	29.3	39.3	894
UniAD-I⊖[10]	36.0	28.9	28.3	2062
DUALAD-I	**49.5**	**34.6**	**55.1**	**663**
DUALAD-I⊖	42.8	31.5	44.4	940

Temporal Consistency, Non-Synchronized Sensors: Since our proposed dual-stream design can flexibly propagate the belief state through time, we run DUALAD in a setting with non-synchronized sensors. To do so, we split the camera images into two sets $\mathcal{C}_{front}$ containing the three front-facing cameras and $\mathcal{C}_{back}$ for the rear cameras respectively. We use the inputs of $\mathcal{C}_{front}$ and $\mathcal{C}_{back}$ in an alternating fashion, leading to three camera inputs per time step and an effective refresh rate per camera of $1\,\text{Hz}$. The resulting performance is shown in Table 7. The restriction of sensor data to one half of the scene per time step decreases the performance for both approaches. We observe that the IDS of UniAD [10] doubles, while DUALAD keeps consistent tracks with only $29\,\%$ increase in IDS. This observation again highlights the effectiveness of the dual-stream design.

Effect on Highly Dynamic Agents: Following our hypothesis that the dual-stream design should especially improve the detection of agents with higher velocities, we con-



(a) DUALAD



(b) DUALVAD

Figure 3. **Qualitative Results**. Fig. 3a shows the output of DUALAD for object tracking, map segmentation, motion prediction and planning. Fig. 3b shows the same scene for the vectorized version DUALVAD of our approach.

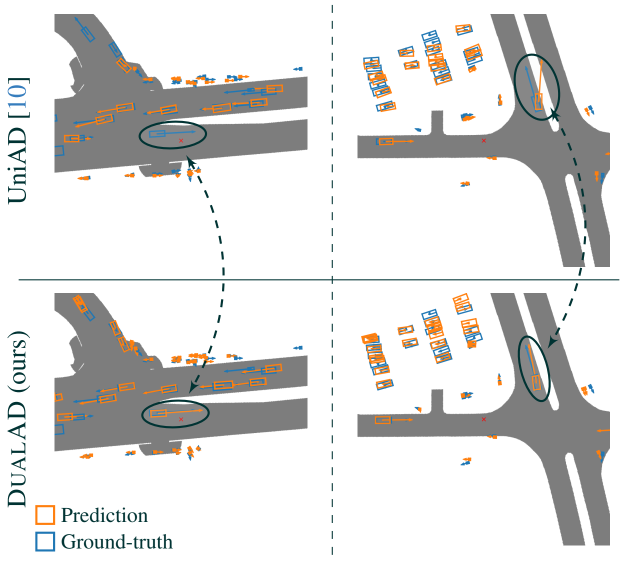


Figure 4. **Performance Comparison** of DUALAD and UniAD [10] for two different scenes. Predictions are shown in orange, ground-truth annotations in blue, ego location with a red cross. While highly dynamic agents cause perception errors such as track losses or distorted objects for UniAD, DUALAD consistently captures them due to the proposed dual-stream design.

duct an experiment in which we focus on highly dynamic agents. More specifically, we evaluate the object detection performance on objects of the type car, where both, the absolute and the relative velocity with respect to the ego vehicle, are higher than $10\,\mathrm{m/s}$. The performance of UniAD [10] drops to 36.3 ($-38\,\%$) mAP while DUALAD drops to 47.3 ($-25\,\%$) mAP, yielding an increased performance delta of 5.5mAP. We observe on the one hand that for both approaches the detection of highly dynamic objects is particularly challenging. On the other hand, these results confirm the importance of explicit motion modelling for the perception of dynamic agents as conducted in DUALAD.

5. Conclusion

This paper presents DUALAD, a novel approach that explicitly models dynamic agents and static scene elements in a dual-stream design, where both can directly access the sensor information. This split explicitly accounts for object and ego motion within the dynamic stream, while only compensating for ego motion within the static stream. The streams can interact by the newly introduced dynamic-static cross-attention, facilitating object detection by utilizing the inferred scene structure around the object.

Our approach not only excels in early-stage perception tasks such as object detection and online map learning, but also demonstrates seamless integration with recent end-to-end models to tackle downstream tasks. In our experimental evaluation, DUALAD yields significant improvements over specialized models and reaches SOTA performance for object detection, map segmentation, and multiple object tracking. Additionally, the integration into end-to-end models revealed improvements in motion prediction and planning, highlighting the importance of our dual-stream design for the entire functional chain.

Whilst our approach results in a robust and temporally consistent perception of the scene, the integration of other modalities such as LiDAR could boost the performance even further, especially combined with the potential of our model to flexibly move the belief state to different points in time to incorporate even unsynchronized sensors. The integration of additional information like traffic signs or traffic lights, as well as the integration of additional tasks such as depth-estimation or lane topology reasoning, remain promising research directions.

Acknowledgement: This work is a result of the joint research project STADT:up (19A22006O). The project is supported by the German Federal Ministry for Economic Affairs and Climate Action (BMWK), based on a decision of the German Bundestag. The author is solely responsible for the content of this publication.

References

[1] 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 *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 4, 5, 6, 1

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 2

[3] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[4] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020. 1

[5] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems (NIPS)*, 2022. 1

[6] Simon Doll, Richard Schulz, Lukas Schneider, Viviane Benzin, Markus Enzweiler, and Hendrik P.A. Lensch. Spatialdetr: Robust scalable transformer-based 3d object detection from multi-view camera images with global cross-sensor attention. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022. 2, 3

[7] Simon Doll, Niklas Hanselmann, Lukas Schneider, Richard Schulz, Markus Enzweiler, and Hendrik P.A. Lensch. Startrack: Latent motion models for end-to-end 3d object tracking with adaptive spatio-temporal appearance representations. *arXiv.org*, arXiv:2306.17602, 2023. 2, 3, 5

[8] Junru Gu, Chenxu Hu, Tianyuan Zhang, Xuanyao Chen, Yilun Wang, Yue Wang, and Hang Zhao. Vip3d: End-to-end visual trajectory prediction via 3d agent queries. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 4, 6

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5, 6, 3

[10] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[11] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. 2023. 1, 2, 3, 4, 5, 6, 7

[12] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. 2019. 5, 1

[13] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 1

[14] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022. 2, 3, 4, 5, 6, 1

[15] Zhiqi Li, Wenhai Wang, Enze Xie, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, Ping Luo, and Tong Lu. Panoptic segformer: Delving deeper into panoptic segmentation with transformers. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3

[16] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction. 2022. 2, 3, 4

[17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5, 1

[18] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022. 2, 3

[19] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. In *Proc. of the International Conf. on Machine learning (ICML)*, 2023. 2

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. 2017. 5

[21] NuScenes Benchmark. nuScenes Detection Task. https://nuscenes.org/object-detection, . Accessed: 2.11.23. 4, 5, 1

[22] NuScenes Benchmark. nuScenes Tracking Task. https://nuscenes.org/tracking, . Accessed: 2.11.23. 4, 1, 3

[23] Ziqi Pang, Jie Li, Pavel Tokmakov, Dian Chen, Sergey Zagoruyko, and Yu-Xiong Wang. Standing between past and future: Spatio-temporal modeling for multi-camera 3d multi-object tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5, 6

[24] LIAN Qing, Tai Wang, Dahua Lin, and Jiangmiao Pang. Dort: Modeling dynamic objects in recurrent for multi-camera 3d object detection and tracking. In *Proc. Conf. on Robot Learning (CoRL)*, pages 3749–3765. PMLR, 2023. 2

[25] Abbas Sadat, Sergio Casas, Mengye Ren, Xinyu Wu, Pranaab Dhawan, and Raquel Urtasun. Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 1

[26] StreamPETR GitHub. https://github.com/exiawsh/StreamPETR. Accessed: 2.11.23. 5, 1

[27] UniAD GitHub. https://github.com/OpenDriveLab/UniAD. Accessed: 2.11.23. 5, 6, 1

[28] VAD GitHub. https://github.com/hustvl/VAD. Accessed: 2.11.23. 2, 5, 6, 1, 3

[29] Qitai Wang, Yuntao Chen, Ziqi Pang, Naiyan Wang, and Zhaoxiang Zhang. Immortal tracker: Tracklet never dies. *arXiv.org*, arXiv:2111.13672, 2021. 2, 6

[30] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. 2023. 2, 3, 5, 6, 7, 1

[31] Xiaofeng Wang, Zheng Zhu, Yunpeng Zhang, Guan Huang, Yun Ye, Wenbo Xu, Ziwei Chen, and Xingang Wang. Are we ready for vision-centric driving streaming perception? the asap benchmark. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 4

[32] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Proc. Conf. on Robot Learning (CoRL)*, 2022. 2, 3

[33] Zitian Wang, Zehao Huang, Jiahui Fu, Naiyan Wang, and Si Liu. Object as query: Lifting any 2d object detector to 3d detection. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 3791–3800, 2023. 2

[34] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 5, 7

[35] Jiang-Tian Zhai, Ze Feng, Jinhao Du, Yongqiang Mao, Jiang-Jiang Liu, Zichang Tan, Yifu Zhang, Xiaoqing Ye, and Jingdong Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv.org*, arXiv:2305.10430, 2023. 6

[36] Tianyuan Zhang, Xuanyao Chen, Yue Wang, Yilun Wang, and Hang Zhao. Mutr3d: A multi-camera tracking framework via 3d-to-2d queries. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 7

[37] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv.org*, arXiv:2205.09743, 2022. 6

[38] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv.org*, arXiv:2010.04159, 2020. 2, 4

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 $18\,\text{GB}$ 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 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**