V-Max: Making RL Practical for Autonomous Driving

Anonymous Authors
Paper under double-blind review

Keywords: mid-to-end; Autonomous Driving; Reinforcement Learning; Framework

Summary

Learning-based decision-making has the potential to enable generalizable Autonomous Driving (AD) policies, reducing the engineering overhead of rule-based approaches. Imitation Learning (IL) remains the dominant paradigm, benefiting from large-scale human demonstration datasets, but it suffers from inherent limitations such as distribution shift and imitation gaps. Reinforcement Learning (RL) presents a promising alternative, yet its adoption in AD remains limited due to the lack of standardized and efficient research frameworks. To this end, we introduce V-Max, an open research framework providing all the necessary tools to make RL practical for AD. V-Max is built on Waymax (Gulino et al., 2023), a hardware-accelerated AD simulator designed for large-scale experimentation. We extend it using ScenarioNet's (Li et al., 2023b) approach, enabling the fast simulation of diverse AD datasets.

Contribution(s)

1. We introduce V-Max, an open research framework to make RL practical for mid-to-end autonomous driving.
   Context: Waymax (Gulino et al., 2023) is an accelerated, data-driven simulator. Hardware-acceleration makes it a compelling simulator to train RL policies, however it requires re-implementing all the elements of the RL pipeline. V-Max does this work, and provides a modular pipeline, including observation functions, encoders, rewards, and algorithms. MetaDrive (Li et al., 2023a) also propose tools to apply RL to the mid-to-end task, but it does not support hardware-acceleration.
2. V-Max enables the simulation of diverse datasets, it also implements various evaluation metrics and enable adversarial evaluation.
   Context: Besides the RL training pipeline, these features aim to make V-Max a standard benchmark for mid-to-end AD. ScenarioNet (Li et al., 2023b) proposes a unified data format for AD, we adapt this approach to make datasets compatible with Waymax, enabling notably for the first time the accelerated simulation of nuPlan (Caesar et al., 2021). We complete the evaluation metrics proposed in Waymax with the ones of the nuPlan benchmark, to provide an unified evaluation score. We include ReGentS (Yin et al., 2024), a gradient-based method that generates adversarial agents, to test the robustness of driving policies.
3. Using V-Max, we conduct an experimental study of design choices in RL for AD. We end up producing highly performing RL agents. We also implement IL and rule-based baselines to show V-Max's versatility.
   Context: We believe to be the first to perform a study of this kind, and that our findings can accelerate RL research on V-Max. We produce a Soft Actor-Critic (SAC, Haarnoja et al. (2018)) agent that solves 97% of the scenarios in the non-reactive evaluation setting, demonstrating that RL can achieve strong performance in this task. However, since there is still no unified evaluation system for the task, we do not claim to be SOTA. In our final benchmark, RL dominates the other approaches, but we did not tune them as much, and did not implement the SOTA of IL. The aim of the benchmark is to show that V-Max can be used with all kind of approaches.
4. We publicly release V-Max and all the components to reproduce our experiments. We detail in the supplementary materials all the hyperparameters needed to reproduce our results.
   Context: None

V-Max: Making RL Practical for Autonomous Driving

Anonymous authors
Paper under double-blind review

Abstract

Learning-based decision-making has the potential to enable generalizable Autonomous Driving (AD) policies, reducing the engineering overhead of rule-based approaches. Imitation Learning (IL) remains the dominant paradigm, benefiting from large-scale human demonstration datasets, but it suffers from inherent limitations such as distribution shift and imitation gaps. Reinforcement Learning (RL) presents a promising alternative, yet its adoption in AD remains limited due to the lack of standardized and efficient research frameworks. To this end, we introduce V-Max, an open research framework providing all the necessary tools to make RL practical for AD. V-Max is built on Waymax (Gulino et al., 2023), a hardware-accelerated AD simulator designed for large-scale experimentation. We extend it using ScenarioNet's (Li et al., 2023b) approach, enabling the fast simulation of diverse AD datasets. V-Max integrates a set of observation and reward functions, transformer-based encoders, and training pipelines. Additionally, it includes adversarial evaluation settings and an extensive set of evaluation metrics. Through a large-scale benchmark, we analyze how network architectures, observation functions, training data, and reward shaping impact RL performance.

Code is available at: ... [1]

1 Introduction

Reinforcement Learning (RL, Sutton & Barto (2018)) has proven to be a powerful approach for controlling real-world systems, with milestones in dexterous robotic manipulation and industrial process control (Rajeswaran et al., 2018; Degrave et al., 2022). RL's ability to learn adaptive policies through closed-loop interaction makes it an appealing framework for Autonomous Driving (AD, Kiran et al. (2022)), where decision-making agents must continuously respond to unseen scenarios and distribution shifts while maintaining high levels of robustness.

However, applying RL to real-world tasks such as AD introduces significant challenges, particularly regarding sample efficiency and training environments. As a result, RL remains underused in AD research due to practical constraints. Imitation Learning (IL, Bansal et al. (2019)) is often favored instead, as it capitalizes on vast driving datasets collected by vehicle fleets and reduces decision-making to a supervised learning task. The absence of RL-compatible environments made RL unusable in the only public challenge for AD (Karnchanachari et al., 2024), which led the organizers to conclude that learning-based methods could not compete with simple rule-based baselines (Dauner et al., 2023).

This gap has motivated recent efforts to improve the accessibility of RL research for AD. Notably, ScenarioNet provides an open-source framework for standardizing and replaying AD datasets in MetaDrive, an RL-compatible simulator that facilitates research on RL generalization in driving (Li et al., 2023a;b). In parallel, Gulino et al. (2023) released Waymax, a hardware-accelerated driving simulator capable of running large-scale simulations at unprecedented speeds, making RL's sample inefficiency less of a limiting factor for experimentation. Waymax was developed as a

[1] Code will be published on GitHub after the double-blind reviewing process, a zipped folder is joined to the submission.

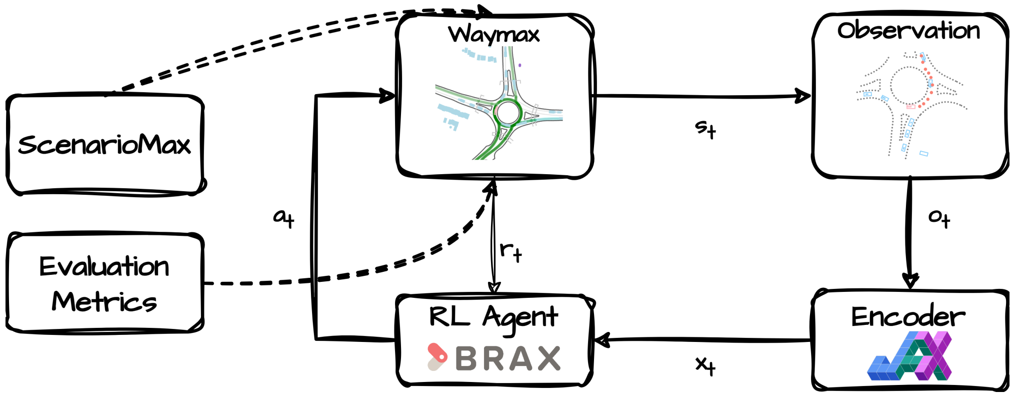


Figure 1: **Overview of the V-Max framework.** ScenarioMax standardizes multiple datasets into a Waymax-compatible format. The simulation runs in Waymax (Gulino et al., 2023), which provides the simulator state s_t. An observation o_t is extracted and processed using a JAX-based neural encoder (Bradbury et al., 2018) before being fed into an RL agent implemented with Brax (Freeman et al., 2021). The RL agent selects an action a_t (acceleration, steering), which is executed in the simulator, receiving a reward r_t based on evaluation metrics. JAX enables to run multiple instances of this process in parallel, on the same device.

high-speed simulation tool, but it lacks essential benchmarking capabilities for RL research, requiring practitioners to build full training pipelines from scratch.

In this work, we introduce V-Max, a framework that extends Waymax with all the necessary tools for RL research in autonomous driving. V-Max provides a set of observation and reward functions, multiple transformer-based encoders, and a complete training pipeline for standard RL algorithms. All these elements are implemented using the JAX framework(Bradbury et al., 2018), enabling training and simulation to be performed within the same computation graph. Additionally, V-Max leverages ScenarioNet's approach to enable the accelerated simulation of diverse driving datasets, whereas Waymax was originally limited to the Waymo Open Motion Dataset (WOMD, Ettinger et al. (2021)). With these features, V-Max aims to standardize RL experimentation for AD, making algorithm comparisons more reproducible and accelerating progress in learning-based decision-making.

We enhance Waymax's evaluation metrics by reimplementing nuPlan's metrics (Karnchanachari et al., 2024) and introducing additional metrics, such as traffic light violations, for a more comprehensive assessment of policy performance. To further evaluate robustness, we integrate *ReGentS* (Yin et al., 2024), enabling evaluation against adversarial agents. We conduct a large-scale benchmark with these tools, systematically analyzing how observation functions, reward shaping, training data selection, network architectures, and learning algorithms impact performance and sample efficiency. These experiments demonstrate V-Max's versatility, facilitating research and development on decision-making for AD.

Our contributions are as follows:

1. V-Max provides a fully integrated, JAX-based, RL training pipeline, including observation and reward functions, and transformer-based encoders inspired by motion forecasting.
2. V-Max supports multi-dataset accelerated simulation by extending Waymax with ScenarioNet's approach.
3. V-Max integrates comprehensive evaluation tools, including the reimplementation of nuPlan's driving quality metrics, and integration of ReGentS for adversarial evaluation.
4. We perform a benchmark on the impact of network architectures, observation choices, reward shaping, and training data on RL performance in AD, resulting in a policy that succesfully completes 97.4% of the scenarios in WOMD.

2 Related Work

2.1 Reinforcement Learning for Autonomous Driving

There are two main formulations of the Autonomous Driving (AD) task in the literature. The first category consists of *end-to-end* approaches (Chen et al., 2024), which aim to learn vehicle controls directly from raw sensor data. Kendall et al. (2019) successfully applied End-to-End RL to lane-following in real-world settings, while Toromanoff et al. (2020) won the first CARLA (Dosovitskiy et al., 2017) challenge using Reinforcement Learning (RL) with a supervised pretraining. These works demonstrated RL's potential in AD, particularly as a way to overcome the limitations of Imitation Learning (IL), such as distribution shift, causal confusion and imitation gap (Walsman et al., 2022). However, methods based solely on RL still fail to perform in the end-to-end setting, the main reason being that RL gradients are insufficient to train the large neural networks needed for perception (Chen et al., 2024). This issue is further compounded by the difficulty of creating realistic and fast simulators for the closed-loop training required for RL. Most works rely on the CARLA simulator, which allows procedurally generated scenarios to be played in the Unreal Engine (Dosovitskiy et al., 2017). While generative world models such as GAIA-1 (Hu et al., 2023) offer photorealistic closed-loop simulation, their computational cost remains a barrier to large-scale RL training.

The parallel approach is to work at mid-level and decouple the decision-making problem from the real-world perception task. In this *mid-to-end* paradigm, agents process post-perception data, i.e. a structured high-level representation of the scene, and output vehicle controls. The release of large post-perception datasets like WOMD, nuScenes and Argoverse 2 (Caesar et al., 2020; Ettinger et al., 2021; Wilson et al., 2021) accelerated mid-to-end research, with a focus on the trajectory prediction sub-task. Closed-loop evaluation and training of mid-level agents was made possible with the appearance of data-driven simulators, that can replay scenarios from real-world driving while taking into account the agent's actions. Research on mid-level decision-making mainly revolves around IL and methods to improve its robustness, such as data augmentation (Bansal et al., 2019), model-based generative adversarial IL (MGAIL) (Bronstein et al., 2022), policy gradients (Scheel et al., 2022), and curriculum learning (Bronstein et al., 2023). Notably, the *nuPlan Challenge 2023* (Karnchanachari et al., 2024) remains the only public competition for the mid-to-end AD task, and its closed-loop challenge was won by PDM (Dauner et al., 2023), a rule-based approach that significantly outperformed all the other learning-based approaches, which were all variants of imitation learning.

Lu et al. (2023) demonstrated that combining IL and RL with a simple reward signal can improve policy robustness in corner cases underrepresented in the training dataset. Similarly, Grislain et al. (2024) showed that incorporating an RL objective is needed to mitigate the imitation gap, which arises from the discrepancy between the observations of human experts and those of mid-to-end AD agents (e.g. sound, turn signals). Cusumano-Towner et al. (2025) showed that self-play can generate highly robust policies, surpassing all prior approaches on CARLA, nuPlan, and Waymax. Their work heavily relies on a proprietary high-speed simulator, highlighting how accelerated simulation can enable large-scale RL training and significantly impact learning-based decision-making for AD.

2.2 Frameworks for mid-to-end Autonomous Driving

V-Max is a framework built on Waymax (Gulino et al., 2023) which is a data-driven, accelerated, mid-to-end AD simulator. Besides Waymax, other frameworks related to V-Max include nuPlan (Caesar et al., 2021), Nocturne (Vinitsky et al., 2023), MetaDrive (Li et al., 2023a), and GPUDrive (Kazemkhani et al., 2024). Below, we compare them to V-Max.

Datasets. All the aforementioned frameworks enable data-driven simulation, where driving scenes are instantiated by replaying real-world data. MetaDrive also integrates procedural generation, allowing to artificially instantiate driving maps and specific situations (e.g. lane merging, roundabouts). Nocturne, GPUDrive and Waymax are limited to the WOMD dataset (Ettinger et al., 2021), while nuPlan uses its own dataset. MetaDrive and V-Max are compatible support both nuPlan and WOMD, as well as other datasets like Argoverse 2 (Wilson et al., 2021), thanks to the use of ScenarioNet's standardization (Li et al., 2023b).

Hardware-Acceleration. Waymax supports both acceleration on GPUs and TPUs enabling high speed simulation. If additionally the training pipeline is written using the JAX library (Bradbury et al., 2018), which is the case in V-Max, then simulation and training can be performed within the same computation graph, eliminating communication bottlenecks with the host machine. GPUDrive achieves GPU-acceleration through the Madrona game engine (Shacklett et al., 2023). Hardware-acceleration makes V-Max, Waymax and GPUDrive two to three orders of magnitude faster than CPU-based simulators like nuPlan, MetaDrive, and Nocturne.

Multi-Agent Environments. Waymax supports environments with multiple controllable agents, a feature that V-Max uses to perform adversarial evaluation. While multi-agent RL (MARL) can technically be implemented in Waymax, V-Max is designed for traditional single-agent RL and does not include MARL-specific functionalities. In contrast, GPUDrive is explicitly designed and optimized for multi-agent learning, making it the better choice for MARL and self-play applications.

Observation. In the mid-to-end setting, simulators provide perfect perception of the scene, making the first design choice the selection of what the driving agent observes. There are two approaches to this decision. The first approach models partial observability to reduce the sim-to-real gap. Nocturne and GPUDrive use sensor-based observations that replicate camera or LiDAR properties, where vehicles can occlude one another. V-Max also implements these sensor-based observations, along with the noisy observations from IGDrivSim (Grislain et al., 2024), which were designed to highlight the limitations of IL. The second approach, observation shaping, focuses on selecting an observation that maximizes policy performance while minimizing memory usage. V-Max provides tools for observation shaping and includes a comparison of different observation choices in Table 2, a topic not addressed in other frameworks.

Evaluation. MetaDrive, Nocturne, Waymax, and GPUDrive evaluate driving agents using a goal-reaching metric, which measures the percentage of scenarios where an agent successfully reaches its destination without collisions or off-road violations. nuPlan introduces a more sophisticated scoring system that also considers driving quality. V-Max integrates both the goal-reaching metric from Waymax and nuPlan's scoring system, enabling more comprehensive evaluations and facilitating direct comparisons between agents.

3 The V-Max Framework

Figure 1 provides an overview of the V-Max framework, which formulates mid-to-end AD as a partially observable Markov Decision Process (POMDP, Spaan (2012)). In this section, we present the core components of V-Max and how they extend Waymax to make RL practical for AD.

3.1 Rules of the Game

Simulation. The simulation process leverages a `simulator_state` that encapsulates data from a bird's-eye view (BEV) representation, under the assumption that the perception problem is fully resolved. This `simulator_state` includes comprehensive records of real-world scenarios, encompassing logged trajectories and high-definition (HD) maps. The primary objective of the ego vehicle is to predict control outputs, specifically acceleration and steering, to govern the vehicle's motion from time t to $t + 1$. Vehicle dynamics are modeled using a continuous bicycle model, which forms the basis for motion planning and control. The simulation operates over a 9-second scenario duration, running at a frequency of 10 Hz. The initial second of each scenario is typically simulated using log-replay to establish a historical context for scene perception.

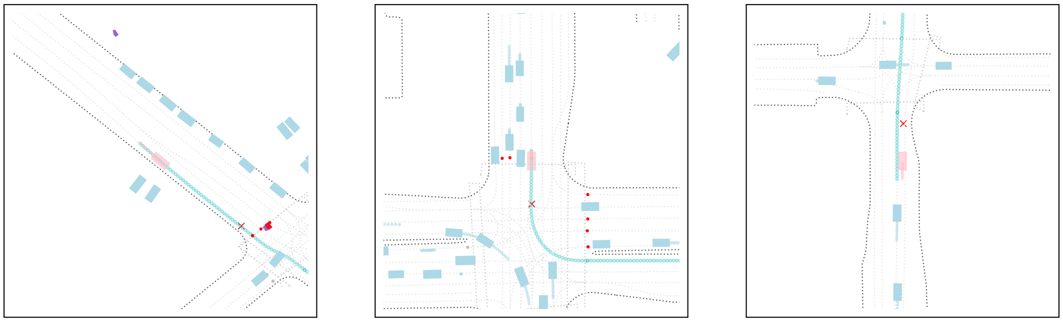


Figure 2: **Illustration of potential limitations when utilizing expert trajectories as learning targets.** The pink rectangle represents the ego, and the blue rectangles are the other vehicules. The blue path is one SDC path and the red cross is the last waypoint of the expert's ground truth. *Left:* The trajectory terminates before a traffic signal, inadvertently encoding the implicit knowledge that the expert stopped at a red light. *Center:* The trajectory ends in the middle of the intersection, showing that the expert stopped to let other vehicles pass, which unintentionally teaches the policy when to yield. *Right:* The trajectory ends immediately prior to an intersection, which may result in the policy incidentally avoiding collisions by terminating at this location. In each scenario, we overlay the self-driving vehicle (SDC) path in blue, which provides a topologically consistent road representation without encoding such implicit behavioral biases, thus constituting a more appropriate supervisory signal for policy optimization.

The scenario concludes when the ego vehicle violates critical safety constraints. Critical failures considered include collisions with other objects, deviations from the road, and crossing intersections under a red light. Notably, the latter constraint is not originally present in the Waymax framework and has been introduced within the V-Max framework.

Goal. V-Max does not prescribe a universal goal for the policy; instead, it allows practitioners to define the desired behavior of the ego vehicle thanks to SDC (Self-Driving Car) paths. Waymax defines SDC path as the routes given to an agent by combining the logged future trajectory of the agent with all possible future routes after the logged trajectories. At the time of writing, these paths are not publicly available in WOMD. An alternative is to rely on expert-logged trajectories only as reference paths. However, this approach is problematic because expert trajectories represent privileged information that consistently demonstrates safe behavior, as shown in Figure 2. To overcome this limitation, V-Max enhances the simulation environment by incorporating reconstructed SDC paths. This addition enables researchers to define various practical tasks such as navigating to specific destinations or following predetermined routes.

3.2 Training RL Agents

ScenarioMax. One of V-Max's key contributions is ScenarioMax, an extension of ScenarioNet (Li et al., 2023b) that converts multiple open-source driving datasets into a single, compatible TfRecord format. This integration process requires several preprocessing steps to ensure data consistency and quality across different sources.

Our approach includes SDC paths reconstruction by creating drivable area definitions for the ego vehicle using road lane data. Since the original SDC paths are not publicly available, we derive them from the simulator state information. We construct paths by starting at the lane closest to the SDC's initial position, then following exit lanes. When multiple lane options exist, we create separate paths. Our method generates 10 distinct paths, selected based on their proximity to the SDC's final position. This approach captures important route options while maintaining diverse targets. Improvements could be made by adding adjacent lanes, allowing for more complex maneuvers such as safe lane changing.

While ScenarioNet proposed a scenario description format, Waymax simulator requires specific data fields to construct the `simulator_state`. To address this gap, we augment the HD map data by adding directional vectors to each map point and defining proper roadgraph types. We also apply proper labeling to match the tf.Example format used by the Waymo Open Motion Dataset (WOMD).

Training pipeline. V-Max uses a flexible wrapper system to encapsulate environments, drawing inspiration from the Brax (Freeman et al., 2021) framework's approach to parallel simulation while extending it for autonomous driving.

Notable wrappers include the AutoResetWrapper that restarts scenarios automatically when completed and the VmapWrapper that handles batched scenarios during training to accelerate policy development. We significantly modified the BraxWrapper to better integrate with our learning processes. We also added a wrapper to reconstruct one SDC path on the fly in a simulator state to support the original WOMD dataset. This wrapper is not fully recommended as it can contains errors due to the difficulty to reconstruct dynamic data in JAX jitted functions.

To support diverse learning paradigms, V-Max provides a standardized training pipeline that creates consistent agent-simulator interactions across different learning approaches (imitation learning, off-policy, and on-policy methods). Observation and feature extraction wrappers provide a flexible mechanism for processing BEV data and state representations. The reward function module is designed for customization, allowing practitioners to define task-specific objectives and shape agent behavior through tailored incentives.

In addition to these foundational components, V-Max includes popular decision-making algorithms implementations, facilitating rapid experimentation with different policy-learning techniques. A dedicated encoder catalog further enhances the system by offering a range of neural network architectures optimized for extracting high-level representations from input features.

Observation function. Selecting the right input features is essential for the performance of learning-based methods in autonomous driving. While Waymax provides a function to transform the simulator state to the self-driving car (SDC) view, it doesn't offer complete tools to build input features for neural networks. To solve this problem, we developed feature extractors that organize data into input features such as: (1) trajectory features showing how object motion; (2) roadgraph features describing roads and lanes; (3) traffic light features showing signal states; and (4) path target features indicating where the vehicle should go. Figure 3 shows how the data is processed from the `simulator_state` to adequate features for a neural network.

The entire feature extraction system can be customized through `yaml` configurations, giving practitioners flexibility in designing observations.

Network architectures. To process mid-level observations, we leverage architectures developed for motion forecasting challenges (Ettinger et al., 2021; Wilson et al., 2021). These challenges focus on predicting the future trajectories of all agents in a driving scene and use the same structured scene representations as our task. Since most motion forecasting models are built on encoder-decoder architectures, their encoders can be repurposed to extract meaningful features from a mid-level driving scene, making them suitable as value and policy networks in RL algorithms.

The motion forecasting competitions are dominated by transformer-based architectures (Vaswani et al., 2017). The attention mechanism is particularly useful for encoding temporal dependencies in the SDC's past trajectory, modeling interactions between the SDC and other road users, and capturing relationships between the SDC and road features. These properties make transformers a compelling architecture for our task. While Waymax reports training results with the Wayformer architecture (Nayakanti et al., 2023), no official public implementation is available. We reimplement Wayformer along with other state-of-the-art encoders from the motion forecasting literature using JAX, enabling in-graph training.

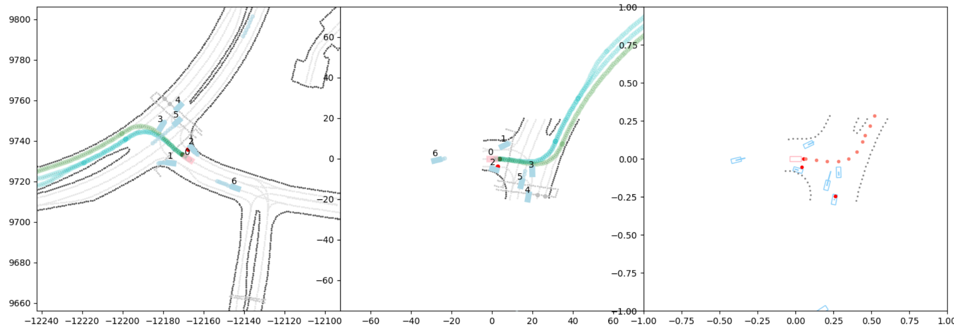


Figure 3: **Visualization of the observation process**. *Left:* Scene-centered view of a scenario. SDC paths are displayed, in blue paths containing the expert trajectory, and in green showing alternative route options. *Center*: Ego-centric transformation with HD map filtering within a rectangular bounding box (70 meters front, 5 meters back, 20 meters on both sides of the SDC). Optionally, noise and masking can be applied to the perception of the scene. *Right:* Neural network input representation. Road boundaries are highlighted after roadgraph filtering. Only the eight closest objects to the SDC are retained, and the SDC path containing the ground truth is selected and interpolated into 10 points spaced 5 meters apart, providing a compact representation of the environment for decision making.

Reward function. In Waymax, the reward is defined as a weighted sum of multiple components. We follow this approach and extend it by adding more reward functions based on the metrics defined in subsection 3.3.

3.3 Evaluation and Benchmarking

Metrics. Waymax proposes the following metrics: collision rate, offroad rate, route progress ratio, and average displacement error (ADE, ℓ_2-distance between the agent's position and the expert's position at each timestep, averaged over the trajectory). We re-implemented the metrics used in the nuPlan challenge (Karnchanachari et al., 2024), which provide a more fine-grained assessment of driving quality. Notably, nuPlan distinguishes between the collisions imputable to the agent's action, and unavoidable incidents, such as rear-end collisions. Additionally, we integrate a red-light violation check, a feature absent from both Waymax and nuPlan.

The WOMD dataset does not provide speed limits, so to compute nuPlan's speed-limit compliance metric, we inferred the speed limit from the expert trajectory using the following methodology. The dataset's roads are located in San Francisco or Phoenix, where speed limits are one of $\{25, 35, 45, 70\}$ mph. Additionally, road metadata indicates whether the agent is on a highway or in an urban scenario. We estimate the speed limit as 70 mph if and only if the agent is on a highway; otherwise, we assign the lowest speed limit such that the expert does not exceed it.

For the comfort metric, which is based on jerk and acceleration values, we initially adopted the same bounds as nuPlan. However, we observed that the ground truth trajectories had unexpectedly poor comfort scores, with only 40% of trajectories classified as comfortable. We identified that the computation of jerk magnitude ($||{}^{d^3\vec{v}}/_{dt^3}||$) produced unrealistic values, leading us to remove it from the metric. With this modification, the expert is classified as comfortable in 82% of WOMD scenarios. Ideally, this percentage should be closer to 100%, indicating that the comfort metric still requires further investigation.

Episode score. To aggregate multiple metrics into a single score, we adopt the methodology from the nuPlan challenge (Karnchanachari et al., 2024). Each episode is assigned a score based on a hybrid weighted average of all metric scores:

$$\text{episode score} = \prod_{i \in \text{multiplier metrics}} \text{score}_i \times \sum_{j \in \text{average metrics}} \text{weight}_j \times \text{score}_j$$

The complete list of metrics and their corresponding weights are provided in the supplementary material.

Evaluations setups. The main evaluation setup used in V-Max is closed-loop non-reactive, where other agents replay their logged trajectories. The advantage of this setup is that all non-controlled agents exhibit human-like behavior, as they follow real-world recorded data. However, a key limitation arises when the agent's actions deviate from those originally taken by the expert, leading to unrealistic interactions. A common example is when an agent drives slower than the expert, causing other vehicles to collide with it from behind. This issue is partly mitigated by the short duration of scenarios (8s) and nuPlan's distinction between at-fault and unavoidable collisions.

Waymax includes a closed-loop reactive evaluation setup, where agents follow their logged trajectories but adjust their speed using an IDM policy (Treiber et al., 2000).By default, all agents continue driving straight once they reach the end of their logged trajectory, regardless of road geometry. Additionally, stationary vehicles (e.g., those stopped at traffic lights or parked for the entire scenario) are initialized at the end of their logged trajectory. This causes them to start moving in a straight line, leading to unrealistic behaviors. These limitations make Waymax's reactive evaluation setup unrealistic, so we chose not to include it in our experiments. nuPlan's reactive agents also use an IDM policy, but use the roadgraph to generate their trajectories, resulting in more realistic behavior. We plan to integrate this feature in a future release.

Another evaluation setup available in V-Max applies Gaussian perturbations to the first 10 timesteps of the agent's trajectory, following the methodology of Bansal et al. (2019). This setup assesses the policy's ability to recover from distribution shifts, as agents in the training data are most often initialized at the center of their lane.

V-Max also integrates ReGentS (Yin et al., 2024), a methodology for generating adversarial scenarios by modifying real-world driving data. In ReGentS, surrounding objects (e.g., vehicles, cyclists, and pedestrians) are optimized to create challenging situations for the agent while maintaining realistic and physically plausible interactions. The method prevents unrealistic swinging turns and unavoidable rear-end collisions, ensuring that the generated scenarios provide meaningful robustness evaluations.

4 Case Study: RL Design Choice for Autonomous Driving

We evaluated V-Max through extensive experiments that demonstrate its ability to replicate and benchmark methodologies. Our experiments include studies examining observation functions, reward formulations, and neural architectures, showing how V-Max enables both reproduction of existing methods and development of new approaches.

All experiments[2] were executed across 3 random seeds, with results presented as means and standard deviations. Accuracy denotes the percentage of episodes completed without failure conditions (collisions, off-road, or traffic signal violations). Additional metrics include collision rate, off-road rate, and route progress ratio—analogous to Waymax's metric. The *V-Max Score* is an extension of the nuPlan score by adding the cross red light metric . All evaluations were performed on the WOMD validation dataset comprising 44,096 distinct scenarios, with a maximum of 64 objects.

We present a control configuration for that serves all of our experiments:

Table 1: Control configuration

Algorithm	Observation	Encoder	Reward	Dataset Training	Dataset Evaluation
SAC	Road	LQ	Navigation	WOMD Training	WOMD Valid

[2]Runs are executed on one single NVIDIA L4 GPU for 12-24 hours per run.

Observation experiments. We implemented four distinct observation functions to explore how different input representations affect driving performance: (1) **Base**: incorporates all available data types for comprehensive scene representation; (2) **Segment**: focuses on road segments with the traffic light present on the target path; (3) **Lane**: includes only lane centers for trajectory guidance; and (4) **Road**: specifically emphasizes road boundaries to define the drivable area. All observation functions maintain consistent representation of key elements: they include the n closest objects, n closest traffic lights, and the same path target definition. This path target consists of a SDC path interpolated to 10 points spaced at 5-meter intervals.

Table 2: Observation study, with control configuration 1.

Observation	Accuracy ↑	Collision ↓	Off-road ↓	Progress ↑	V-Max Score ↑
Base	96.92±0.20	1.94±0.18	0.91±0.09	**173.32±4.21**	0.85±0.00
Segment	96.15±0.35	1.80±0.29	1.83±0.22	152.85±15.12	0.84±0.01
Lane	95.99±0.42	2.18±0.41	1.60±0.06	136.67±10.79	0.84±0.00
Road	**97.26±0.28**	**1.76±0.18**	**0.83±0.07**	165.46±3.21	**0.86±0.01**

Network encoders. We provide to the users of V-Max a catalog of several encoder architectures, implemented with the Flax library (Heek et al., 2024): (1) **Latent-query** (LQ): inspired from (Jaegle et al., 2021), (2) **Latent-query hierarchical** (LQH) (Bronstein et al., 2022); (3) **Motion Transformer** (MTR) (Shi et al., 2024), (4) **Wayformer** (Nayakanti et al., 2023). For comparison, we also take an architecture that uses one multi-layer percepetron to encode each feature (road, trajectories...) separately: **MLP**. And an architecture that don't use seperate encodings: **None**.

The results of Table 3 clearly demonstrate the substantial impact of encoder selection on overall performance. The Latent-query (LQ) encoder achieves the best results across all metrics, while other transformer-based architectures (LQH, MTR, and Wayformer) perform similarly. The MLP encoder shows considerably worse results, and the baseline "None" condition performs extremely poorly. These findings highlight the critical importance of transformer-based encoders for effective scene understanding in autonomous driving.

Table 3: Encoder architectures study, with control configuration 1.

Encoder	Accuracy ↑	Collision ↓	Off-road ↓	Progress ↑	V-Max Score ↑
None	69.95±1.72	25.13±1.40	4.51±0.23	87.26±3.43	0.53±0.01
MLP	87.54±0.48	9.24±0.31	2.92±0.24	104.03±2.93	0.68±0.01
LQ	**97.26±0.28**	**1.76±0.18**	**0.83±0.07**	**165.46±3.21**	**0.86±0.01**
LQH	96.28±0.35	2.52±0.12	1.02±0.20	162.23±12.26	0.84±0.01
MTR	95.94±0.24	2.42±0.24	1.42±0.38	154.88±2.53	0.84±0.01
Wayformer	96.08±0.42	2.70±0.41	0.99±0.11	161.94±7.20	0.84±0.00

Reward shaping. Tuning the weights of the various reward components to achieve the best possible policy remains a challenging task. To address this, we conducted a comprehensive benchmark of different reward functions by calibrating these parameters: the choice of metrics included in the reward and the weights assigned to each metric.

$$r_{\text{safety}}(s_t, a_t) = -\mathbb{I}[\text{Collided}] - \mathbb{I}[\text{Off-road}] - \mathbb{I}[\text{Red light crossed}] \qquad \textbf{(Safety)}$$

$$r_{\text{navigation}}(s_t, a_t) = r_{\text{safety}}(s_t, a_t) - 0.6 \cdot \mathbb{I}[\text{Offroute}] + 0.2 \cdot \mathbb{I}[\text{Progressed}] \qquad \textbf{(Navigation)}$$

$$r_{\text{behavior}}(s_t, a_t) = r_{\text{navigation}}(s_t, a_t) - 0.3 \cdot \mathbb{I}[\text{Speed}] - 0.3 \cdot \mathbb{I}[\text{TTC}] + 0.5 \cdot \mathbb{I}[\text{Comfort}] \qquad \textbf{(Behavior)}$$

Findings in Table 4 indicate that the Navigation reward function provides the optimal balance between safety and route completion, suggesting that more complex reward structures may introduce competing objectives. This highlights the critical role of reward design in developing autonomous driving policies that effectively balance safety and driving efficiency.

Table 4: Reward shaping study, with control configuration 1

Reward	Accuracy ↑	Collision ↓	Off-road ↓	Progress ↑	V-Max Score ↑
Safety	96.73±0.57	2.21±0.29	0.90±0.34	78.66±3.18	0.67±0.04
Navigation	**97.26±0.28**	**1.76±0.18**	**0.83±0.07**	165.46±3.21	**0.86±0.01**
Behavior	96.17±0.29	2.47±0.20	1.12±0.06	**199.84±4.03**	0.83±0.02

Cross-dataset experiments. To evaluate the effectiveness of *ScenarioMax*, we tested three training approaches: using WOMD alone, using nuPlan alone, and combining both datasets. Table 5 shows the performance results across all validation datasets. While we initially expected the combined training to consistently outperform single-dataset training in all scenarios, the results show that performance levels are actually quite similar. Nevertheless, the mixed-data policy shows a key advantage: it maintains good performance across different validation sets, demonstrating better generalization, while policies trained on individual datasets perform worse when tested on other data distributions.

Table 5: Cross-datasets study, with control configuration 1.

Dataset	Accuracy ↑	Collision ↓	Off-road ↓	Progress ↑	V-Max Score ↑
Evaluated on WOMD dataset					
WOMD	**97.26±0.28**	**1.76±0.18**	**0.83±0.07**	165.46±3.21	**0.86±0.01**
nuPlan	76.81±1.53	7.30±1.25	1.30±0.28	163.77±8.15	0.67±0.01
MIX	96.24±0.68	2.49±0.57	0.98±0.10	**172.29±10.28**	0.85±0.02
Evaluated on nuPlan dataset					
WOMD	87.73±0.64	**1.89±0.20**	2.66±0.15	308.64±4.04	0.76±0.01
nuPlan	95.33±0.42	2.27±0.23	1.86±0.17	**319.84±14.02**	**0.82±0.00**
MIX	**95.38±0.71**	2.04±0.40	**1.98±0.21**	315.95±15.77	**0.82±0.01**
Evaluated on MIX dataset					
WOMD	94.21±0.05	**1.80±0.13**	1.42±0.07	211.43±3.18	0.83±0.01
nuPlan	82.76±1.16	5.69±0.92	1.47±0.22	213.79±9.95	0.72±0.01
MIX	**95.97±0.68**	2.35±0.51	**1.29±0.13**	**218.33±11.96**	**0.84±0.01**

5 Benchmark

Building on the insights from Section 4, where we explored the versatility of the V-Max framework in terms of observation functions, reward functions, network encoders, and multi-dataset training, we now shift our focus to evaluating the performance and robustness of reinforcement learning algorithms. In this section, we examine populars planning algorithms and assess their effectiveness under various evaluation scenarios. In this section, the result of the best performing model is reported.

5.1 Planning algorithms

The first methodology is evaluating planning methods on standard non-reactive (NR) evaluation. The standard policies included *expert*, *random*, and *constant*, while the rule-based policies consisted of *IDM* and *PDM* (Dauner et al., 2023). For the learning-based policies, we tested four algorithms from both IL and RL: *BC*, *PPO* (Schulman et al., 2017), *SAC* (Haarnoja et al., 2018), and *BC_SAC* (Lu et al., 2023). It is important to note that we did not fine-tune the training hyper-parameters for the learning-based methods. The reported results reflect their performance under default or standard settings, rather than an optimized configuration. Table 6 displays the best results obtained from methods available in V-Max.

Table 6: Benchmarking planning algorithms, with control configuration 1.

Planning Policies	Accuracy ↑	Collision ↓	Off-road ↓	Progress ↑	V-Max Score ↑
Expert	98.06	0.56	0.76	97.30	0.93
Constant	55.26	27.56	11.34	87.67	0.51
Random	12.60	34.40	38.40	82.40	0.10
IDM	88.20	7.50	3.80	151.00	0.81
PDM[†]	93.40	4.70	1.40	158.00	0.82
BC (discrete)	79.42	13.14	6.92	86.87	0.72
PPO	90.75	7.81	1.14	**189.52**	0.78
SAC	**97.44**	**1.74**	**0.74**	169.01	**0.88**
BC_SAC	96.61	2.16	1.04	159.61	0.86

5.2 Robustness analysis

To thoroughly assess the robustness of our best-performing planning method, we designed and executed two distinct experimental setups: initialization perturbation and adversarial attacks.

Initialization perturbation As described in Section 3, we compare our top-performing RL model, BC model and the rule-based PDM method with initialization perturbation. These evaluations were performed on a smaller validation dataset consisting of 294 scenarios. The results in Table 7 demonstrates that RL can adapt to initial noise and dynamically re-center itself to the correct lane, whereas imitation method struggle since they rigidly mimic demonstrations without the ability to recover from disturbances.

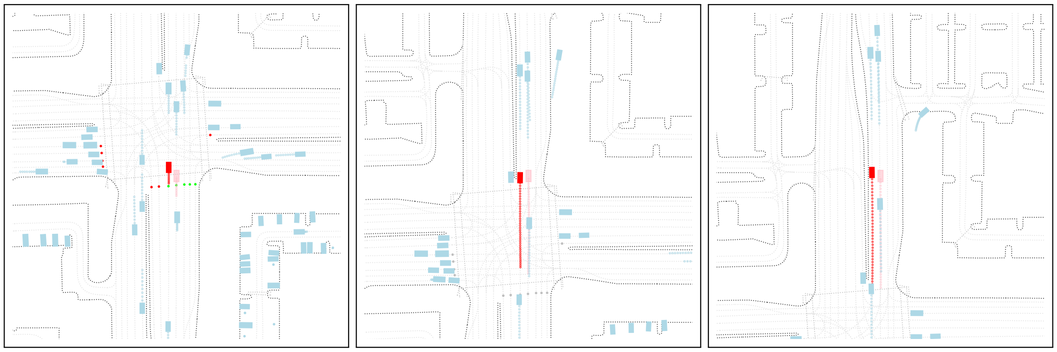
Table 7: Benchmarking evaluations methods, results on the first 294 scenarios of WOMD valid, with control configuration 1.

Algorithm	Evaluation	Accuracy ↑	Collision ↓	Offroad ↓	Progress ↑	V-Max score ↑
SAC	Non-reactive	97.40	1.70	0.74	169.01	0.88
	Noise Init	94.50	3.00	1.70	162.29	0.83
BC	Non-reactive	84.64	8.87	5.8	90.18	0.75
	Noise Init	35.15	32.76	27.64	52.65	0.25
PDM[†]	Non-reactive	93.50	4.40	1.70	152.17	0.82
	Noise Init	91.5	5.5	2.4	158	0.79

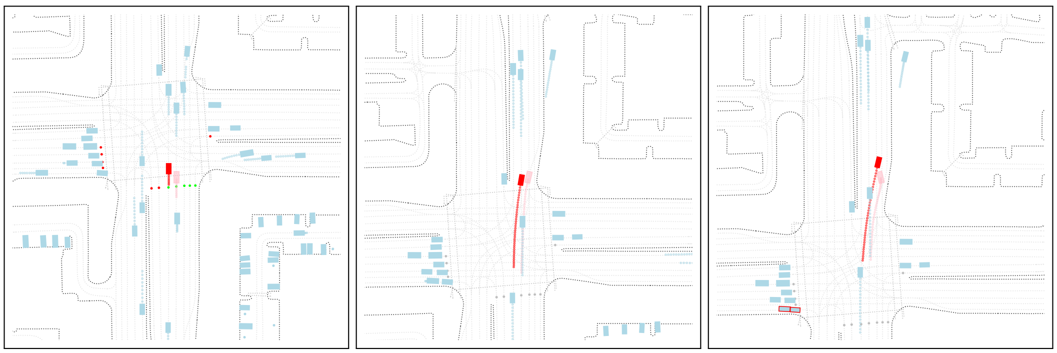
Adversarial attack We also investigated how our best RL agent performs under adversarial attacks. However, evaluating this process is challenging, as the methodology is not universally applicable to all scenarios (red light stop, no close surrounding objects) and requires extensive tuning to ensure a rigorous and scientifically robust assessment. To explore this further, we applied the ReGentS process to a selected set of scenarios. The results of this evaluation are presented in the Figure 4 displaying the adversarial process on one episode.

6 Conclusion

In this work, we introduced V-Max, a framework designed to make Reinforcement Learning (RL) practical for mid-to-end Autonomous Driving (AD). Built on Waymax, V-Max extends its capabilities with a JAX-based RL training pipeline, multi-dataset accelerated simulation, and comprehensive



(a) Scene-centered view of the initial scenario without applying adversarial attack.



(b) Scene-centered view after applying ReGentS. The adversarial agent's (in red) trajectory to collide with SDC agent. We can observe RL agent's adaptation with the path deviation to avoid collision and re-centering itself on the lane to follow after adversarial objects passes.

Figure 4: **Visualization of the ReGentS process.** Comparison between a standard and an adversarial scenario.

evaluation tools. Using these tools, we trained high-performing SAC agents, showing how V-Max can help advance RL research for AD. To further support progress in this area, we ensure full reproducibility by publishing our framework and benchmarks.

While V-Max provides a foundation for AD research, rigorously evaluating driving policies remains an open challenge. Current evaluation protocols (in V-Max and the frameworks discussed in section 2) average scenario metrics across the entire validation dataset. However, driving difficulty follows a long-tail distribution (Makansi et al., 2021; Bronstein et al., 2023), where most scenarios are easily solvable while a small subset presents significant challenges. Developing benchmarks that explicitly account for this distribution would enable a more rigorous assessment of policy robustness.

Additionally, further research on adversarial scenario generation, could enable deeper robustness assessment of driving policies. ReGentS is a good starting point, diffuser-based methods could be an alternative approach (Pronovost et al., 2023). Similarly, the development of more realistic simulation agents, as explored in the *Waymo Open Sim Agents Challenge* (WOSAC, (Montali et al., 2023)) could improve realism of closed-loop simulators, reducing the reliance on non-reactive evaluation.

References

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst. In *Proceedings of Robotics: Science and Systems*, volume 15, June 2019. ISBN 978-0-9923747-5-4. URL https://www.roboticsproceedings.org/rss15/p31.html.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

Eli Bronstein, Mark Palatucci, Dominik Notz, Brandyn White, Alex Kuefler, Yiren Lu, Supratik Paul, Payam Nikdel, Paul Mougin, Hongge Chen, Justin Fu, Austin Abrams, Punit Shah, Evan Racah, Benjamin Frenkel, Shimon Whiteson, and Dragomir Anguelov. Hierarchical Model-Based Imitation Learning for Planning in Autonomous Driving. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8652–8659, October 2022. DOI: 10.1109/IROS47612.2022.9981695. URL https://ieeexplore.ieee.org/document/9981695. ISSN: 2153-0866.

Eli Bronstein, Sirish Srinivasan, Supratik Paul, Aman Sinha, Matthew O'Kelly, Payam Nikdel, and Shimon Whiteson. Embedding Synthetic Off-Policy Experience for Autonomous Driving via Zero-Shot Curricula. In *Proceedings of The 6th Conference on Robot Learning*, pp. 188–198. PMLR, March 2023. URL https://proceedings.mlr.press/v205/bronstein23a.html. ISSN: 2640-3498.

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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11618–11628, Seattle, WA, USA, June 2020. IEEE. ISBN 978-1-72817-168-5. DOI: 10.1109/CVPR42600.2020.01164. URL https://ieeexplore.ieee.org/document/9156412/.

Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles, June 2021. URL http://arxiv.org/abs/2106.11810. arXiv:2106.11810 [cs].

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end Autonomous Driving: Challenges and Frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2024. ISSN 1939-3539. DOI: 10.1109/TPAMI.2024.3435937. URL https://ieeexplore.ieee.org/document/10614862/?arnumber=10614862. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Marco Cusumano-Towner, David Hafner, Alex Hertzberg, Brody Huval, Aleksei Petrenko, Eugene Vinitsky, Erik Wijmans, Taylor Killian, Stuart Bowers, Ozan Sener, Philipp Krähenbühl, and Vladlen Koltun. Robust Autonomy Emerges from Self-Play, February 2025. URL http://arxiv.org/abs/2502.03349. arXiv:2502.03349 [cs].

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with Misconceptions about Learning-based Vehicle Motion Planning. In *Proceedings of The 7th Conference on Robot Learning*, pp. 1268–1281. PMLR, December 2023. URL https://proceedings.mlr.press/v229/dauner23a.html. ISSN: 2640-3498.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de las Casas, Craig Donner, Leslie Fritz, Cristian Galperti, Andrea Huber, James Keeling, Maria Tsimpoukelli, Jackie Kay, Antoine Merle, Jean-Marc Moret, Seb Noury, Federico Pesamosca, David Pfau, Olivier Sauter, Cristian Sommariva, Stefano Coda, Basil Duval, Ambrogio Fasoli, Pushmeet Kohli, Koray Kavukcuoglu, Demis Hassabis, and Martin Riedmiller. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, February 2022. ISSN 1476-4687. DOI: 10.1038/s41586-021-04301-9. URL https://www.nature.com/articles/s41586-021-04301-9. Publisher: Nature Publishing Group.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16. PMLR, October 2017. URL https://proceedings.mlr.press/v78/dosovitskiy17a.html. ISSN: 2640-3498.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R. Qi, Yin Zhou, Zoey Yang, Aurélien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large Scale Interactive Motion Forecasting for Autonomous Driving: The Waymo Open Motion Dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9710–9719, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Ettinger_Large_Scale_Interactive_Motion_Forecasting_for_Autonomous_Driving_The_Waymo_ICCV_2021_paper.html.

C. Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax - A Differentiable Physics Engine for Large Scale Rigid Body Simulation, 2021. URL http://github.com/google/brax.

Clémence Grislain, Risto Vuorio, Cong Lu, and Shimon Whiteson. IGDrivSim: A Benchmark for the Imitation Gap in Autonomous Driving, November 2024. URL http://arxiv.org/abs/2411.04653. arXiv:2411.04653 [cs].

Cole Gulino, Justin Fu, Wenjie Luo, George Tucker, Eli Bronstein, Yiren Lu, Jean Harb, Xinlei Pan, Yan Wang, Xiangyu Chen, John Co-Reyes, Rishabh Agarwal, Rebecca Roelofs, Yao Lu, Nico Montali, Paul Mougin, Zoey Yang, Brandyn White, Aleksandra Faust, Rowan McAllister, Dragomir Anguelov, and Benjamin Sapp. Waymax: An Accelerated, Data-Driven Simulator for Large-Scale Autonomous Driving Research. *Advances in Neural Information Processing Systems*, 36:7730–7742, December 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1838feeb71c4b4ea524d0df2f7074245-Abstract-Datasets_and_Benchmarks.html.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1861–1870. PMLR, July 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html. ISSN: 2640-3498.

Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2024. URL http://github.com/google/flax.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A Generative World Model for Autonomous Driving, September 2023. URL http://arxiv.org/abs/2309.17080. arXiv:2309.17080 [cs].

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General Perception with Iterative Attention. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 4651–4664. PMLR, July 2021. URL https://proceedings.mlr.press/v139/jaegle21a.html. ISSN: 2640-3498.

Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, and Holger Caesar. Towards learning-based planning: The nuPlan benchmark for real-world autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 629–636, May 2024. DOI: 10.1109/ICRA57147.2024.10610077. URL https://ieeexplore.ieee.org/document/10610077/?arnumber=10610077.

Saman Kazemkhani, Aarav Pandya, Daphne Cornelisse, Brennan Shacklett, and Eugene Vinitsky. GPUDrive: Data-driven, multi-agent driving simulation at 1 million FPS. In *The Thirteenth International Conference on Learning Representations*, October 2024. URL https://openreview.net/forum?id=ERv8ptegFi.

Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to Drive in a Day. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8248–8254, Montreal, QC, Canada, May 2019. IEEE. ISBN 978-1-5386-6027-0. DOI: 10.1109/ICRA.2019.8793742. URL https://ieeexplore.ieee.org/document/8793742/.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep Reinforcement Learning for Autonomous Driving: A Survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, June 2022. ISSN 1558-0016. DOI: 10.1109/TITS.2021.3054625. URL https://ieeexplore.ieee.org/abstract/document/9351818. Conference Name: IEEE Transactions on Intelligent Transportation Systems.

Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. MetaDrive: Composing Diverse Driving Scenarios for Generalizable Reinforcement Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3461–3475, March 2023a. ISSN 1939-3539. DOI: 10.1109/TPAMI.2022.3190471. URL https://ieeexplore.ieee.org/document/9829243. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Quanyi Li, Zhenghao (Mark) Peng, Lan Feng, Zhizheng Liu, Chenda Duan, Wenjie Mo, and Bolei Zhou. ScenarioNet: Open-Source Platform for Large-Scale Traffic Scenario Simulation and Modeling. *Advances in Neural Information Processing Systems*, 36:3894–3920, December 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/0c26a501df8fb919a0350e2df06b5d39-Abstract-Datasets_and_Benchmarks.html.

Yiren Lu, Justin Fu, George Tucker, Xinlei Pan, Eli Bronstein, Rebecca Roelofs, Benjamin Sapp, Brandyn White, Aleksandra Faust, Shimon Whiteson, Dragomir Anguelov, and Sergey Levine. Imitation Is Not Enough: Robustifying Imitation with Reinforcement Learning for Challenging Driving Scenarios. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7553–7560, October 2023. DOI: 10.1109/IROS55552.2023.10342038. ISSN: 2153-0866.

Osama Makansi, Özgün Çiçek, Yassine Marrakchi, and Thomas Brox. On Exposing the Challenging Long Tail in Future Prediction of Traffic Actors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13147–13157, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Makansi_On_Exposing_the_Challenging_Long_Tail_in_Future_Prediction_of_ICCV_2021_paper.html.

Nico Montali, John Lambert, Paul Mougin, Alex Kuefler, Nicholas Rhinehart, Michelle Li, Cole Gulino, Tristan Emrich, Zoey Zeyu Yang, Shimon Whiteson, Brandyn White, and Dragomir Anguelov. The Waymo Open Sim Agents Challenge. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=5FnttJZQFn.

Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S. Refaat, and Benjamin Sapp. Wayformer: Motion Forecasting via Simple & Efficient Attention Networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2980–2987, London, United Kingdom, May 2023. IEEE. ISBN 9798350323658. DOI: 10.1109/ICRA48891.2023.10160609. URL https://ieeexplore.ieee.org/document/10160609/.

Ethan Pronovost, Meghana Reddy Ganesina, Noureldin Hendy, Zeyu Wang, Andres Morales, Kai Wang, and Nick Roy. Scenario Diffusion: Controllable Driving Scenario Generation With Diffusion. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 68873–68894. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d95cb79a3421e6d9b6c9a9008c4d07c5-Paper-Conference.pdf.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Robotics: Science and Systems XIV*, June 2018. URL https://www.roboticsproceedings.org/rss14/p49.html.

Oliver Scheel, Luca Bergamini, Maciej Wolczyk, Błażej Osiński, and Peter Ondruska. Urban Driver: Learning to Drive from Real-world Demonstrations Using Policy Gradients. In *Proceedings of the 5th Conference on Robot Learning*, pp. 718–728. PMLR, January 2022. URL https://proceedings.mlr.press/v164/scheel22a.html. ISSN: 2640-3498.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017. URL http://arxiv.org/abs/1707.06347. arXiv:1707.06347 [cs].

Brennan Shacklett, Luc Guy Rosenzweig, Zhiqiang Xie, Bidipta Sarkar, Andrew Szot, Erik Wijmans, Vladlen Koltun, Dhruv Batra, and Kayvon Fatahalian. An Extensible, Data-Oriented Architecture for High-Performance, Many-World Simulation. *ACM Trans. Graph.*, 42(4), 2023. DOI: 10.1145/3592427.

Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. MTR++: Multi-Agent Motion Prediction With Symmetric Scene Modeling and Guided Intention Querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3955–3971, May 2024. ISSN 1939-3539. DOI: 10.1109/TPAMI.2024.3352811. URL https://ieeexplore-ieee-org.minesparis-psl.idm.oclc.org/document/10398503. 1 citations (Crossref) [2024-06-06] Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Matthijs T. J. Spaan. Partially Observable Markov Decision Processes. In Marco Wiering and Martijn van Otterlo (eds.), *Reinforcement Learning: State of the Art*, pp. 387–414. Springer Verlag, 2012.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning, second edition: An Introduction*. Bradford Books, Cambridge, Massachusetts, 2nd edition edition, November 2018. ISBN 978-0-262-03924-6. URL http://incompleteideas.net/book/the-book-2nd.html.

Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. End-to-End Model-Free Reinforcement Learning for Urban Driving Using Implicit Affordances. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7153–7162, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Toromanoff_End-to-End_Model-Free_Reinforcement_Learning_for_Urban_Driving_Using_Implicit_Affordances_CVPR_2020_paper.html.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824, August 2000. ISSN 1063-651X, 1095-3787. DOI: 10.1103/PhysRevE.62.1805. URL https://link.aps.org/doi/10.1103/PhysRevE.62.1805.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Eugene Vinitsky, Nathan Lichtlé, Xiaomeng Yang, Brandon Amos, and Jakob Foerster. Nocturne: a scalable driving benchmark for bringing multi-agent learning one step closer to the real world, February 2023. URL http://arxiv.org/abs/2206.09889. arXiv:2206.09889 [cs].

Aaron Walsman, Muru Zhang, Sanjiban Choudhury, Dieter Fox, and Ali Farhadi. Impossibly Good Experts and How to Follow Them. In *The Eleventh International Conference on Learning Representations*, September 2022. URL https://openreview.net/forum?id=sciA_xgYofB.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next Generation Datasets for Self-Driving Perception and Forecasting. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 1, December 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/4734ba6f3de83d861c3176a6273cac6d-Abstract-round2.html.

Yuan Yin, Pegah Khayatan, Eloi Zablocki, Alexandre Boulch, and Matthieu Cord. ReGentS: Real-World Safety-Critical Driving Scenario Generation Made Stable. In *ECCV 2024 Workshop on Multimodal Perception and Comprehension of Corner Cases in Autonomous Driving*, September 2024. URL https://openreview.net/forum?id=dJqcdUgEdw.

Supplementary Materials

The following content was not necessarily subject to peer review.

A Metrics catalog

A.1 Waymax Metrics

- Offroad: Binary flag indicating whether the SDC left the drivable area at any point in the scenario.
- Collision (overlap): Binary flag indicating whether the SDC collided with another object at any point in the scenario.
- Wrongway: Waymax-specific metric based on SDC paths, indicating whether the SDC is more than 3.5 meters away from the closest SDC path.
- Offroute: Similar to wrongway but with respect to on-route paths, which are the SDC paths that contain the expert's logged trajectory.
- `sdc_progress`: computes how much the SDC progressed along on-route paths, and divides it by the distance the expert did cover on those paths. Can be greater than 1.

For example, if the SDC takes a right turn at an intersection while the expert proceeded straight, the SDC will be considered offroute but not wrongway.

A.2 nuPlan Metrics

- Progress along route: same definition as Waymax, but capped to 1.
- Making progress: binary flag indicating if progress along route is superior to 20%.
- At-fault collision: Binary flag following nuPlan's criteria for assigning collision responsibility:
  - Collisions with stopped vehicles are always at-fault.
  - If the SDC is stopped, it is never at-fault.
  - If the SDC is occupying multiple lanes, it is at-fault.
  - Rear-bumper collisions are not at-fault, while front-bumper collisions are at-fault.
- TTC within bound: indicates if the time-to-colllision (ttc) with ahead vehicles remain superior to 0.95s.
- Speed limit compliance: defined by nuPlan as:

$$\text{nuplan\_speed\_compliance} = \max\left(0.0, 1.0 - \frac{\sum_t \text{speed\_violation}_t \cdot \Delta t}{\max(\text{threshold}, 1e-3) \cdot T}\right), \tag{1}$$

  where speed_violation_t is the magnitude of overspeeding at timestep t, Δt is the time step duration, and T is the total scenario duration.
- Driving direction compliance: Based on distance traveled into oncoming traffic. We check if the vehicle is effectively driving into oncoming traffic lanes using the road information, rather than SDC paths.
  - Score = 1.0 if wrong-way distance ≤ 2.0m.
  - Score = 0.5 if wrong-way distance is between 2.0m and 6.0m.
  - Score = 0 if wrong-way distance > 6.0m.
- Comfort: binary indicating if the trajectory is comfortable based on jerk, acceleration and yaw rates.

A.3 V-Max Metrics

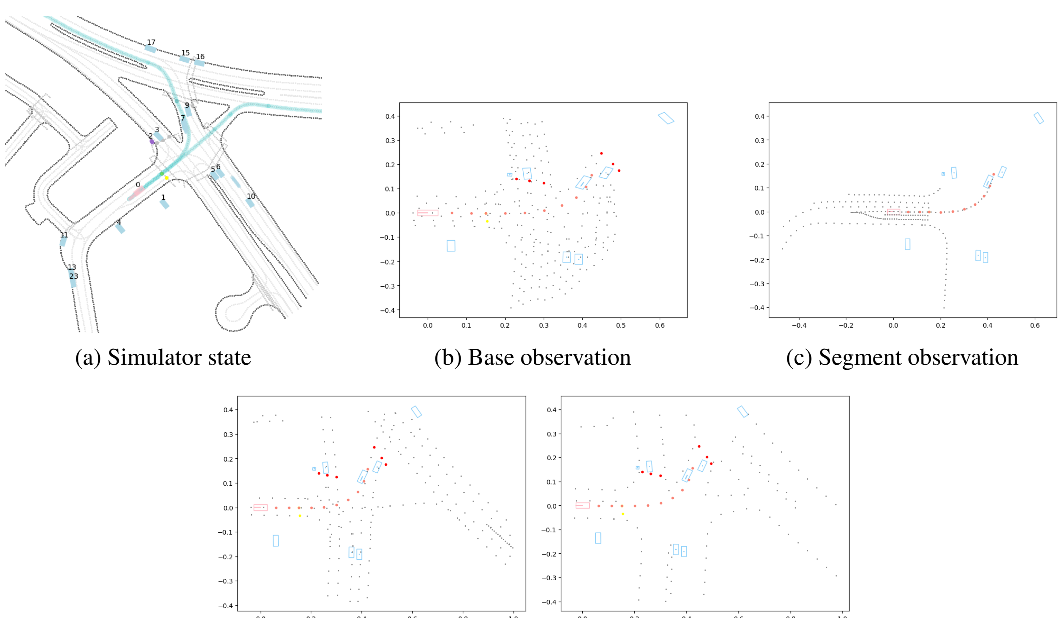
- Red-light violation: Binary flag indicating whether the SDC crossed an intersection while the traffic light was red.
- Time spent on multiple lanes: Evaluated based on roadgraph information rather than SDC paths. We added this metric to encourage agent to remain on one lane, to set the thresholds, we looked at the expert trajectories.
  - Score = 1.0 if time spent on multiple lanes ≤ 3.4s.
  - Score = 0.5 if time spent on multiple lanes is between 3.4s and 5.7s.
  - Score = 0 if time spent on multiple lanes > 5.7s.

Table 8: Metrics and their weights in nuPlan aggregate score and V-Max aggregate score. † indicates metrics that appear only in the V-Max aggregate score.

Metric name	Multiplier weight	Average weight
No at-fault collisions	$\{0, 1\}$	-
Offroad	$\{0, 1\}$	-
Red-light violation †	$\{0, 1\}$	-
Making progress	$\{0, 1\}$	-
Driving direction compliance	$\{0, 0.5, 1\}$	-
TTC within bound	-	5
Progress along route ratio	-	5
Speed limit compliance	-	4
Multiple lanes score†	-	3
Comfort	-	2

B Observation functions

Figure 5: Observation functions illustrated



(a) Simulator state (b) Base observation (c) Segment observation

(d) Lane observation (e) Road observation

C Observation Configurations

Table 9: Observation configurations used in experiments for the **Base** function.

Parameter	Value	Description
`obs_past_num_steps`	5	Number of past steps included in observation
Object Features		
`features`	waypoints, velocity, yaw, size, valid	Object features included in observation
`num_closest_objects`	8	Number of closest objects to consider
Roadgraph Features		
`features`	waypoints, direction, types, valid	Roadgraph features included
`interval`	2	Sampling interval for waypoints
`max_meters`	50	Maximum distance of roadgraph features
`roadgraph_top_k`	200	Top K roadgraph elements
`meters_box`	front: 50, back: 5, left: 20, right: 20	Observation bounding box dimensions in meters
Traffic Light Features		
`features`	waypoints, state, valid	Traffic light features included
`num_closest_traffic_lights`	5	Number of closest traffic lights
Path Target Features		
`features`	waypoints	Target path features included
`num_points`	10	Number of target path points
`points_gap`	5	Gap between target path points

D Training hyperparameters

Table 10: Observation configurations used in experiments for the **Road** function.

Parameter	Value	Description
`obs_past_num_steps`	5	Number of past steps included in observation
Object Features		
`features`	waypoints, velocity, yaw, size, valid	Object features included in observation
`num_closest_objects`	8	Number of closest objects to consider
Roadgraph Features		
`features`	waypoints, direction, valid	Roadgraph features included
`interval`	2	Sampling interval for waypoints
`max_meters`	70	Maximum distance of roadgraph features
`roadgraph_top_k`	200	Top K roadgraph elements
`meters_box`	front: 70, back: 5, left: 20, right: 20	Observation bounding box dimensions in meters
Traffic Light Features		
`features`	waypoints, state, valid	Traffic light features included
`num_closest_traffic_lights`	5	Number of closest traffic lights
Path Target Features		
`features`	waypoints	Target path features included
`num_points`	10	Number of target path points
`points_gap`	5	Gap between target path points

Table 11: Observation configurations used in experiments for the **Lane** function.

Parameter	Value	Description
`obs_past_num_steps`	5	Number of past steps included in observation
Object Features		
`features`	waypoints, velocity, yaw, size, valid	Object features included in observation
`num_closest_objects`	8	Number of closest objects to consider
Roadgraph Features		
`features`	waypoints, direction, valid	Roadgraph features included
`interval`	2	Sampling interval for waypoints
`max_meters`	70	Maximum distance of roadgraph features
`roadgraph_top_k`	300	Top K roadgraph elements
`meters_box`	front: 70, back: 5, left: 20, right: 20	Observation bounding box dimensions in meters
Traffic Light Features		
`features`	waypoints, state, valid	Traffic light features included
`num_closest_traffic_lights`	5	Number of closest traffic lights
Path Target Features		
`features`	waypoints	Target path features included
`num_points`	10	Number of target path points
`points_gap`	5	Gap between target path points

Table 12: Observation configurations used in experiments for the **Segment** function.

Parameter	Value	Description
`obs_past_num_steps`	5	Number of past steps included in observation
Object Features		
`features`	waypoints, velocity, yaw, size, valid	Object features included in observation
`num_closest_objects`	8	Number of closest objects to consider
Roadgraph Features		
`features`	waypoints, direction, types, valid	Roadgraph features included in observation
`max_meters`	50	Maximum distance of roadgraph features
`meters_box`	front: 50, back: 5, left: 20, right: 20	Observation bounding box dimensions in meters
`max_num_lanes`	10	Maximum number of lanes
`max_num_points_per_lane`	20	Maximum points per lane
Traffic Light Features		
`features`	waypoints, state, valid	Traffic light features included
Path Target Features		
`features`	waypoints	Target path features included
`num_points`	10	Number of target path points
`points_gap`	5	Gap between target path points

Table 13: Algorithms hyperparameters used in experiments

Behavioral Cloning (BC)		
Hyperparameter	Value	Description
`Total Timesteps`	200M	Total environment steps done during training
`Learning Rate`	1e-4	The step size for optimization
`Batch Size`	64	Number of samples per gradient update
`Grad updates per steps`	32	Number of gradients backprop per steps
`Loss function`	Log_prob	Cross entropy loss
SAC		
`Total Timesteps`	25M	Total environment steps done during training
`Learning Rate`	1e-4	The step size for optimization
`Batch Size`	64	Number of samples per gradient update
`Discount Factor`	0.99	Discount factor for future rewards
`Entropy rate` α	0.2	Entropy factor for exploration
`Grad updates per steps`	8	Number of gradients backprop per steps
`Buffer size`	1_000_000	Size of the replay buffer
`Learning start`	50_000	Number of random actions to prefill the replay buffer
BC SAC		
`Total Timesteps`	25M	Total environment steps done during training
`Imitation frequency`	8	Frequency where we apply imitation loss instead of RL loss
`RL Learning Rate`	1e-4	The RL learning rate
`Imitation Learning Rate`	5e-5	The IL learning rate
`Batch Size`	64	Number of samples per gradient update
`Discount Factor`	0.99	Discount factor for future rewards
`Entropy rate` α	0.2	Entropy factor for exploration
`Grad updates per steps`	8	Number of gradients backprop per steps
`Buffer size`	1_000_000	Size of the replay buffer
`Learning start`	50_000	Number of random actions to prefill the replay buffer
PPO		
`Total Timesteps`	200M	Total environment steps done during training
`Learning Rate`	1e-4	The step size for optimization
`Batch Size`	512	Number of samples per gradient update
`Num minibatches`	16	Number of sub samples per gradient update
`Discount Factor`	0.99	Discount factor for future rewards
`Entropy rate` α	0.2	Entropy factor for exploration
`Grad updates per steps`	4	Number of gradients backprop per steps
`GAE factor`	0.95	GAE factor for loss computation
`Clip factor` ϵ	0.2	Clipping factor

Table 14: Encoders and decoders hyperparameters used in experiments

MLP policy decoder		
Hyperparameter	Value	Description
`Layer sizes`	[256, 64, 32]	Number and size of layers
`SAC activation fn`	relu	non-linear activation function
`PPO activation fn`	tanh	non-linear activation function
`Parametric action distribution RL`	NormalTanh	Action distribution for RL
`Parametric action distribution BC`	Softmax	Action distribution for IL
`IL activation fn continuous`	tanh	non-linear activation function
MLP value decoder		
`Layer sizes`	[256, 64, 32]	Number and size of layers
`SAC activation fn`	relu	non-linear activation function
`PPO activation fn`	tanh	non-linear activation function
MGAIL encoder		
`Embedding sizes`	[256,256]	Number and size of embedding layers
`dk`	64	dimensionality features of dense encoders
`num latents`	16	size of the learnable latent entry
`cross num heads`	2	number of attention heads
`cross head features`	16	number of features for each attention head
`ff mult`	2	features multiplicator for the feedforward layer size
Perceiver encoder		
`Embedding sizes`	[256,256]	Number and size of embedding layers
`depth`	4	Number of attention layers in the loop
`num latents`	16	size of the learnable latent entry
`num self heads`	2	number of self attention heads
`self head features`	16	number of features for each self attention head
`cross num heads`	2	number of cross attention heads
`cross head features`	16	number of features for each cross attention head
`ff mult`	2	features multiplicator for the feedforward layer size
MTR encoder		
`Embedding sizes`	[256,256]	Number and size of embedding layers
`dk`	64	dimensionality features of dense encoders
`num latents`	16	size of the learnable latent entry
`num self heads`	2	number of self attention heads
`self head features`	16	number of features for each self attention head
`ff mult`	2	features multiplicator for the feedforward layer size
`k`	8	number of nearest objects in attention mechanism
Wayformer encoder		
`Embedding sizes`	[256,256]	Number and size of embedding layers
`dk`	64	dimensionality features of dense encoders
`num latents`	16	size of the learnable latent entry
`num self heads`	2	number of self attention heads
`self head features`	16	number of features for each self attention head
`depth`	2	Number of attention layers in the loop
`ff mult`	2	features multiplicator for the feedforward layer size
`fusion type`	late	late, early or hierarchical fusion attention mechanism