Dense reinforcement learning for safety validation of autonomous vehicles

https://doi.org/10.1038/s41586-023-05732-2

Received: 1 March 2022

Accepted: 16 January 2023

Published online: 22 March 2023

Check for updates

Shuo Feng[1,2,4], Haowei Sun[1], Xintao Yan[1], Haojie Zhu[1], Zhengxia Zou[1,5], Shengyin Shen[2] & Henry X. Liu[1,2,3] ✉

One critical bottleneck that impedes the development and deployment of autonomous vehicles is the prohibitively high economic and time costs required to validate their safety in a naturalistic driving environment, owing to the rarity of safety-critical events[1]. Here we report the development of an intelligent testing environment, where artificial-intelligence-based background agents are trained to validate the safety performances of autonomous vehicles in an accelerated mode, without loss of unbiasedness. From naturalistic driving data, the background agents learn what adversarial manoeuvre to execute through a dense deep-reinforcement-learning (D2RL) approach, in which Markov decision processes are edited by removing non-safety-critical states and reconnecting critical ones so that the information in the training data is densified. D2RL enables neural networks to learn from densified information with safety-critical events and achieves tasks that are intractable for traditional deep-reinforcement-learning approaches. We demonstrate the effectiveness of our approach by testing a highly automated vehicle in both highway and urban test tracks with an augmented-reality environment, combining simulated background vehicles with physical road infrastructure and a real autonomous test vehicle. Our results show that the D2RL-trained agents can accelerate the evaluation process by multiple orders of magnitude (10^3 to 10^5 times faster). In addition, D2RL will enable accelerated testing and training with other safety-critical autonomous systems.

Owing to the rapid development of autonomous vehicle (AV) technologies, we are on the cusp of a revolution in transportation on a scale not seen since the introduction of automobiles a century ago. AV technologies have the potential to substantially improve transportation safety, mobility and sustainability, and thus have attracted worldwide attention from industries, government agencies, professional organizations and academic institutions. Over the past 20 years, substantial progress has been made on the development of AVs, particularly with the emergence of deep learning[2]. By 2015, several companies had announced that they would be mass-producing AVs before 2020[3–5]. So far, the reality has not lived up to these expectations, and no level 4 (ref. [6]) AVs are commercially available. The reasons for this are numerous. But above all, the safety performance of AVs is still substantially below that of human drivers. For average drivers in the United States, the occurrence probability of a crash is around 1.9×10^{-6} per mile in the naturalistic driving environment (NDE)[1]. In contrast, the disengagement rate for the state-of-the-art AV is around 2.0×10^{-5} per mile, according to the 2021 Disengagement Reports from California[7]. Although the disengagement rate is criticized for its potential biasedness, it has been widely used to track the trend of AV safety performance[8,9], as it is arguably the only statistic that is available to the public for the comparison of different AVs.

One critical bottleneck to improving AV safety performance is the severe inefficiency of safety validation. Prevailing approaches usually test AVs in the NDE through a combination of software simulation, closed test track and on-road testing. However, to validate the safety performance of AVs at the level of human drivers, it is well known that hundreds of millions of miles and sometimes hundreds of billions of miles would need to be tested in the NDE[1]. Owing to this severe inefficiency, AV developers must pay substantial economic and time costs to evaluate each development, which has hindered the progress of AV deployment. To improve the testing efficiency, many approaches test AVs in purposely generated scenarios that are more safety critical[10,11]. Yet, existing scenario-based approaches[12–17] can mainly be applied to short scenario segments with limited background road users (see Supplementary Information for more discussions).

Validating the safety performance of AVs in the NDE is in essence a rare-event estimation problem in a high-dimensional space. The main challenge is caused by the compounding effects of the 'curse of rarity' in addition to the 'curse of dimensionality' (Fig. 1a). By 'curse of dimensionality', we mean that driving environments could be spatiotemporally complex, and the variables needed to define such environments are high-dimensional. As the volume of the variable space grows exponentially with dimensionality, the computational complexity also grows exponentially[18]. By 'curse of rarity', we mean that the occurrence probability for safety-critical events is rare, that is, most points of the variable space are non-safety-critical, which provide no or noisy information for

[1]Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI, USA. [2]University of Michigan Transportation Research Institute, Ann Arbor, MI, USA. [3]Mcity, University of Michigan, Ann Arbor, MI, USA. [4]Present address: Department of Automation, Tsinghua University, Beijing, China. [5]Present address: School of Astronautics, Beihang University, Beijing, China. ✉e-mail: henryliu@umich.edu

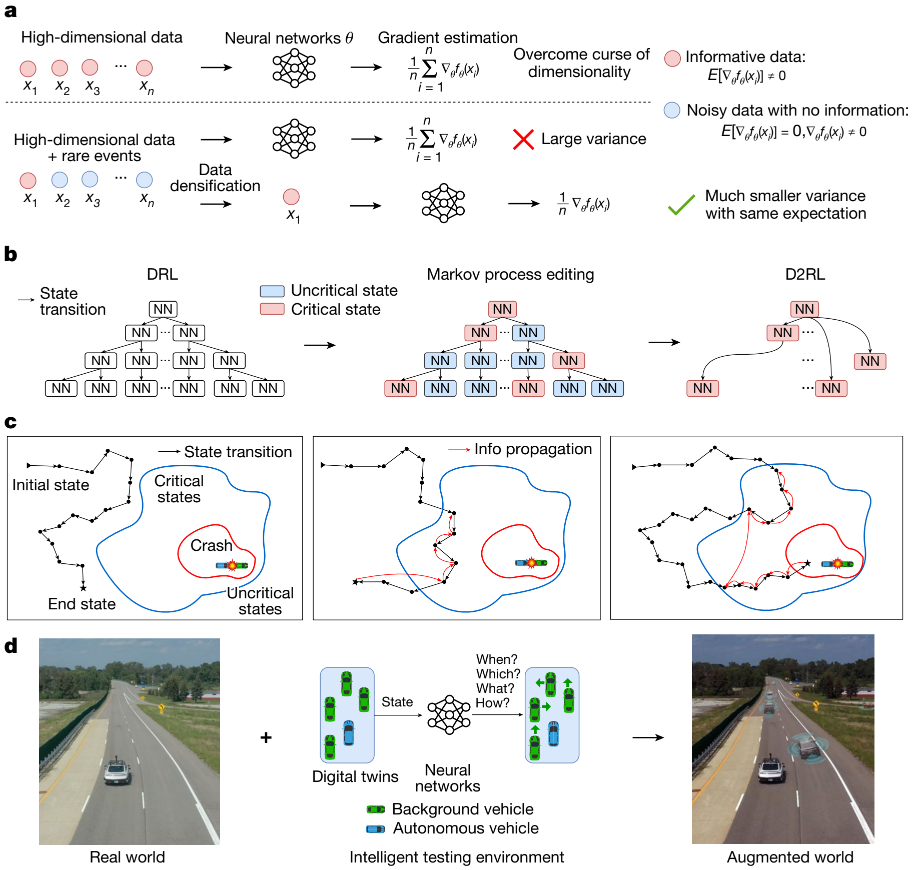
a
High-dimensional data
Neural networks θ
Gradient estimation
x_1 x_2 x_3 ... x_n
$\frac{1}{n}\sum_{i=1}^{n}\nabla_\theta f_\theta(x_i)$
Overcome curse of dimensionality
Informative data: $E[\nabla_\theta f_\theta(x_i)] \neq 0$
Noisy data with no information: $E[\nabla_\theta f_\theta(x_i)] = 0, \nabla_\theta f_\theta(x_i) \neq 0$
High-dimensional data + rare events
Large variance
Data densification
$\frac{1}{n}\nabla_\theta f_\theta(x_i)$
Much smaller variance with same expectation
b
State transition
DRL
Uncritical state
Critical state
Markov process editing
D2RL
NN
c
State transition
Initial state
Critical states
Crash
End state
Uncritical states
Info propagation
d
State
When? Which? What? How?
Digital twins
Neural networks
Background vehicle
Autonomous vehicle
Real world
Intelligent testing environment
Augmented world

Fig. 1 | Validating safety-critical AI with the dense-learning approach. **a**, The curse of rarity hinders the applicability of deep-learning techniques for safety-critical systems, as the gradient estimation of neural networks would suffer from the large variance due to the rareness of informative data. By training the neural networks with the informative data only, our dense-learning approach substantially reduces the gradient estimation variance, enabling deep-learning applications in safety-critical systems. *f* and *E* denote the objective function and mathematical expectation, respectively. **b**, The D2RL approach edits the Markov process by removing the uncritical states and reconnecting the critical states, and then trains the neural networks (NN) for only the edited Markov process. **c**, For any D2RL training episode, the reward from the end state is backpropagated along the edited Markov chain with critical states only. Three examples are provided. In the left example, the episode is completely removed from training data as it does not contain any critical state. In the middle and right examples, the uncritical states are skipped and critical states are reconnected to densify the training data. The end state for the middle example is from a non-crash episode, whereas the right example is from a crash episode. **d**, The augmented-reality testing platform can augment the real world with virtual background traffic, resulting in a safer, more controllable and more efficient testing environment for AVs. Our approach learns to decide when to control which background vehicles to execute what adversarial manoeuvre with what probability.

training. Under this circumstance, it is hard for a deep-learning model to learn even given a large amount of data, as valuable information (for example, policy gradient) of safety-critical events could be buried under the large amount of non-safety-critical data. Recent decades have seen rapid progress in the ability of artificial intelligence (AI) systems to solve problems with the curse of dimensionality[19], for example, the board game Go has a state space of 10^{360} (ref. [20]) and the semiconductor chip design may have a state space on the order of $10^{2,500}$ (ref. [21]). Before this work, however, solving the curse of dimensionality and the curse of rarity simultaneously has remained an open question, which has impeded the applicability of AI techniques in safety-critical systems, such as AVs, medical robots and aerospace systems[22].

We address this challenge by developing a dense deep-reinforcement-learning (D2RL) approach. The basic idea is to identify and remove the non-safety-critical data and train neural networks utilizing the safety-critical data. As only a very small portion of data is safety critical, the information of the remaining data will be substantially densified. Essentially, the D2RL approach edits the Markov decision process by removing the uncritical states and reconnecting the critical states, and then trains neural networks for only the edited Markov process (Fig. 1b). Therefore, for any training episode, the reward from the end state is backpropagated along the edited Markov chain with critical states only (Fig. 1c). The D2RL approach can dramatically reduce the variance of the policy gradient estimation with multiple orders of magnitude without loss of unbiasedness, compared with the DRL approach, as proved in Theorem 1 in Methods. Such substantial variance reduction can enable neural networks to learn and achieve tasks that are intractable for the DRL approach. For AV testing, we leverage the D2RL approach and train the background vehicles (BVs) through a neural network to learn when to execute what adversarial manoeuvre, which aims to improve the testing efficiency and ensure evaluation unbiasedness. This results in an AI-based adversarial testing environment that can reduce the required testing miles of AVs by multiple orders of magnitude while ensuring the testing unbiasedness. Our approach can be applied to complex driving environments, including multiple highways, intersections and roundabouts, which cannot be achieved by previous scenario-based approaches. The proposed approach empowers the testing agents in the environment with intelligence to create an intelligent testing environment, that is, using AI to validate AI. This is a paradigm shift and it opens the door for accelerated testing and training with other safety-critical systems.

To demonstrate the effectiveness of our AI-based testing approach, we trained the BVs with large-scale naturalistic driving datasets and conducted simulation experiments as well as field experiments in physical test tracks. Specifically, we tested a level 4 AV with an open-source automated driving system, Autoware[23], in the physical 4-km-long highway test track at the American Center for Mobility (ACM) and the urban test track at Mcity. To test the AV with the D2RL-trained testing environment safely and precisely, we developed an augmented-reality testing platform[24], which combines the physical test track and a microscopic traffic simulator, SUMO (Simulation of Urban Mobility)[25]. As shown in Fig. 1d, by synchronizing the movements of the real AV and virtual BVs, the real AV in the physical test track can interact with the virtual BVs as though it is in a realistic traffic environment, where the BVs are directed to interact with the real AV. For both simulation and field experiments, we evaluated not only crash rates but also crash types and crash severities. Our simulation and field-testing results show that the D2RL approach can effectively learn the intelligent testing environment, which can substantially accelerate the evaluation process of AVs by multiple orders of magnitude (10^3 to 10^5 times faster) unbiasedly, compared with the results from testing AVs directly in the NDE.

Dense deep reinforcement learning

To leverage AI techniques, we formulate the AV testing problem as a sequential Markov decision process (MDP), where manoeuvres of BVs are decided based on the current state information. We aim to train a policy (a DRL agent) modelled by a neural network, which can control the manoeuvres of BVs to interact with the AV, to maximize the evaluation efficiency and ensure unbiasedness. However, as mentioned earlier, it is hard—or even empirically infeasible—to learn an effective policy if directly applying DRL approaches because of the curse of dimensionality and the curse of rarity.

We address this challenge by developing the D2RL approach. Owing to the rarity of safety-critical events, most states are uncritical and cannot provide information for safety-critical events, so the key concept of D2RL is to remove the data of these uncritical states and utilize only the informative data for training the neural network (Fig. 1b,c). For AV testing problems, many safety metrics[26] can be utilized to identify the critical states with different efficiency and effectiveness. In this study, we utilize the criticality measure[12,13], which is an outer approximation of the AV crash rate within a specific time horizon (for example, one second) from the current state. Theoretical analysis for more generic problems can be found in Methods and Supplementary Section 2a. We then edit the Markov process, discard the data of uncritical states, and use the remaining data for the policy gradient estimation and bootstrapping of the DRL training. We find that dense learning can markedly reduce the variance of the policy-gradient estimation with multiple orders of magnitude without loss of estimation unbiasedness, as proved in Theorem 1 in Methods. The dense learning can also reduce the bootstrapping variance, as it can be regarded as a state-dependent temporal-difference learning[27], where only critical states are utilized and others are skipped.

To demonstrate the effectiveness of dense learning, we compared D2RL with the DRL approach for a corner-case-generation problem[28,29], which can be formulated as a well defined reinforcement-learning problem. A neural network was trained to maximize the AV's crash rate by controlling the closest eight BVs' actions (Fig. 2a). We used proximal policy optimization (PPO)[30] to update the parameters of the policy network, given the reward for each testing episode, that is, +20 for an AV crash and 0 for others. For a fair comparison, the only difference between DRL and D2RL is that DRL utilized all the data for training the neural network, whereas D2RL utilized only the data of critical states. As shown in Fig. 2b, D2RL removed the data of 80.5% complete episodes and 99.3% steps from uncritical states, compared with DRL. According to Theorem 1, this indicates that D2RL can reduce around 99.3% of the policy-gradient-estimation variance, which enables the neural network to learn effectively. Specifically, the D2RL can maximize the reward during the training process, whereas the DRL was stuck from the beginning of the training process (Fig. 2c). The policy learned by D2RL can effectively increase the crash rate of the AV, whereas DRL failed to do so (Fig. 2d). Figure 2e–g illustrates three generated corner cases.

Learning the intelligent testing environment

Learning the intelligent testing environment for unbiased and efficient AV evaluation is much more complex than corner-case generation. According to the importance sampling theory[31], the goal is essentially to learn new sampling distributions, that is, the importance function, of BVs' manoeuvres to replace their naturalistic ones, with the aim of minimizing the estimation variance of AV testing. Intuitively, the BVs are trained to learn when to execute what adversarial manoeuvre, in that all BVs follow naturalistic behaviours, only selected vehicles at selected moments execute specifically designed adversarial moves with a learned probability. To achieve this goal, without using any heuristics or handcrafted functions, we derive the reward function from the estimation variance as

$$r(\mathbf{x}) = -\mathbb{I}_A(\mathbf{x}) \times W_{q_\pi}(\mathbf{x}) \times W_{q_{\pi_\mathrm{b}}}(\mathbf{x}), \qquad (1)$$

where $\mathbf{x}$ denotes the variables of each testing episode, $\mathbb{I}_A(\mathbf{x})$ is an indicator function of the AV crash event (A), and $W_{q_\pi}(\mathbf{x}) = P(\mathbf{x})/q_\pi(\mathbf{x})$ and $W_{q_{\pi_\mathrm{b}}}(\mathbf{x}) = P(\mathbf{x})/q_{\pi_\mathrm{b}}(\mathbf{x})$ are weights (or likelihoods) produced by importance sampling. Here $P(\mathbf{x})$ denotes the naturalistic distribution, $q_\pi(\mathbf{x})$ denotes the importance function with the target policy π, and $q_{\pi_\mathrm{b}}(\mathbf{x})$ denotes the importance function with the behaviour policy π_b. As there is no heuristic or handcrafted immediate reward function, the reward function in equation (1) is highly consistent with the testing performance, that is, a higher reward indicates a more efficient testing environment. Such reward design is generic and applicable to other rare-event estimation problems with high-dimensional variables.

To determine the learning mechanism, we further investigate the relationship between the behaviour policy π_b and target policy π. As proved in Theorem 2 in Methods, we find that the optimal behaviour policy π_b^* that collects data during the training process is nearly inversely proportional to the target policy. It indicates that, if using on-policy learning mechanisms ($q_{\pi_b} = q_\pi$), the behaviour policy would be far from optimality, which could mislead the training process and eventually cause the underestimation issues. To address this issue, we design an off-policy learning mechanism, where a generic behaviour policy is designed and kept unchanged during the training process. Although this off-policy mechanism is not the optimal behaviour policy as in Theorem 2 (which is usually unavailable in practice), it can balance the exploration and exploitation and is empirically effective for all experiment settings in this study. With the reward function and off-policy learning mechanism, we can learn the intelligent testing environment by the D2RL approach (see Methods for training details).

AV testing in simulation

We evaluated the effectiveness of the D2RL-based intelligent testing environment regarding accuracy, efficiency, scalability and generalizability by systematic simulation analysis. To measure the safety performance of AVs, crash rates of different crash types and severities in the NDE are utilized as the benchmark. As the NDE is generated completely based on naturalistic driving data, testing results in the NDE can represent the safety performance of AVs in the real world. For each test episode, we simulated AV driving in traffic for a fixed distance, and then the test results were recorded and analysed. To investigate the scalability and generalizability, we conducted simulation experiments with different road geometries, different driving distances and two different types of AV model (that is, the AV-I and AV-II models; see Supplementary Section 3d).

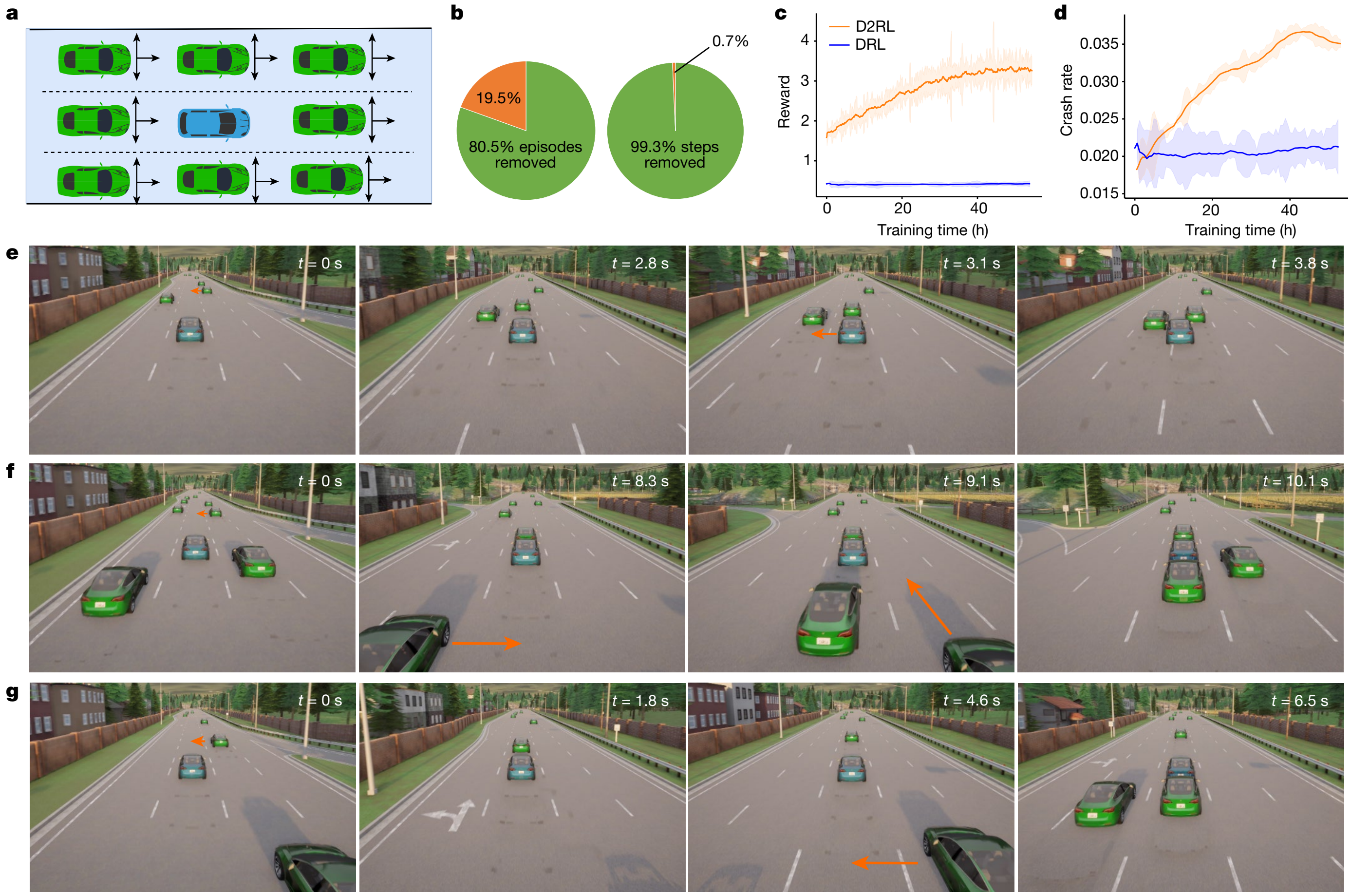

Fig. 2 | Comparison of D2RL with DRL using the corner-case-generation examples. a, The neural network controls the closest 8 vehicles' manoeuvres within 120 m, where each BV has 33 discrete actions at every 0.1 s: left lane change, 31 discrete longitudinal accelerations ([−4, 2] with 0.2 m s^{-2} discrete resolution) and right lane change. **b**, Proportions of the removed data by D2RL regarding the episodes (left) and steps (right). **c**,**d**, Comparison of training rewards between DRL and D2RL (**c**) and comparison of crash rates between the policies learned by DRL and D2RL (**d**). The solid lines represent the moving averages of rewards (**c**) and crash rates (**d**), and the shaded areas represent the standard deviation. **e**, The AV (blue vehicle) made an evasive lane change to avoid a cut-in vehicle but collided with an adjacent vehicle. **f**, The right-front vehicle made a cut-in, the left-behind vehicle made a right lane change, while the right-behind vehicle accelerated. These three vehicles cooperatively encircled the AV and caused a crash. **g**, The right-front vehicle made a cut-in to enforce the AV for braking, which created the opportunity for the right-behind vehicle to make a lane change after 2.8 s (that is, 28 uncritical steps), leading to a crash. Additional explanations are provided in Supplementary Video 1.

Figure 3 shows the results of the two-lane highway environment with the 400-m driving distance for the AV-I model, which is a basic experiment to validate our approach. As shown in Fig. 3a, during the training process, the estimation variance of the intelligent testing environment decreases with the increase of reward function, which demonstrates the effectiveness of the reward function in equation (1). To justify the off-policy mechanism, we investigated the performance of the on-policy mechanism, where the target policy was utilized as the behaviour policy. As shown in Fig. 3b, during the training process, the crash rate for the on-policy experiments substantially increases, whereas the crash rate for the off-policy experiments is unchanged because the behaviour policy is unchanged. However, as the on-policy mechanism breaks the consistency between the reward function and estimation variance, this increase of the crash rate would be misleading. As shown in Fig. 3c, the testing environment obtained by the on-policy mechanism underestimates the crash rate. In contrast, our off-policy approach can obtain the same crash rate as the NDE approach, but more efficiently (Fig. 3d,e). To measure the efficiency, we calculated the minimum number of tests for reaching a predetermined precision threshold (the relative half-width[12,17] is 0.3). To reduce the randomness of the results for a fair comparison, we repeated the testing of our approach by bootstrap sampling and obtained the frequency and average of the required number of tests (Fig. 3f). Compared with the NDE approach that required 1.9 × 10^8 number of tests, our approach required an average of 9.1 × 10^4 number of tests, which is 2.1 × 10^3 times faster. To investigate the generalizability, we further tested the AV-II model using the same intelligent testing environment without any refinement, which can also obtain an accurate estimation with about 10^3 times faster (see Supplementary Section 4d).

To validate the unbiasedness about crash types, crash severities and near-miss events, we analysed the crash rates of different crash types, distribution of the speed difference at the crash moment, and distributions of the time to collision, bumper-to-bumper distance and post-encroachment time of near-miss events. Throughout the paper, our use of the term unbiasedness refers to the fact that estimations from our approach have the same mathematical expectations as those from the NDE. In our experiments, we collected about 2.34 × 10^8 episodes of tests in the NDE and 3.15 × 10^6 (about two orders of magnitude less) episodes of tests in the intelligent testing environment. As the intelligent testing environment is more adversarial than the NDE, the total crash rate in our approach is 3.21 × 10^{-3} (Fig. 3g), which is much higher than that (1.58 × 10^{-7}) in the NDE. As required by the importance sampling theory, each crash event should be weighted by the likelihood ratio to keep the unbiasedness. Therefore, the weighted crash rates for all crash types are compared with

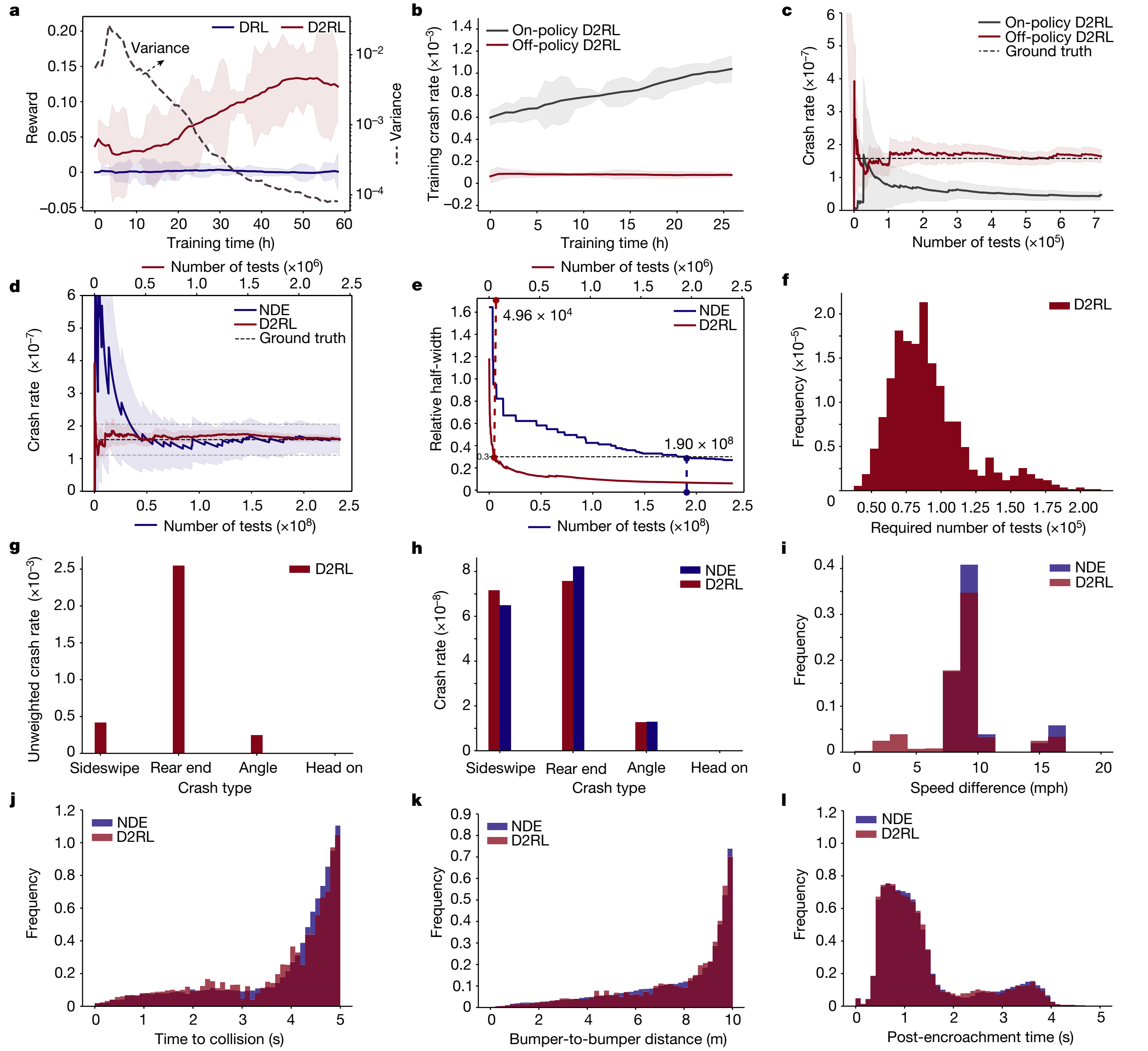

Fig. 3 | Performance evaluation of the D2RL-based intelligent testing environment. a, Comparison of the reward between the DRL and D2RL approaches, along with the estimation variance (dashed line) of the D2RL approach that represents the testing efficiency. The solid lines represent the moving average and the shaded areas represent the standard deviation. **b**, Comparison of crash rates of the on-policy and off-policy D2RL approaches during the training process (**b**) and comparison of estimated crash rates of the on-policy and off-policy D2RL approaches during the testing process (**c**). The shaded area represents the 90% confidence level and the solid lines represent the averages. **d**,**e**, Crash rate estimations (**d**) and relative half-width (**e**) of the AV-I model by the NDE and the D2RL-based intelligent testing environment. The bottom *x* axis denotes the number of tests for the NDE and the top *x* axis denotes the number of tests for the intelligent testing environment. The shaded area represents the 90% confidence level and the solid lines represent the averages (**d**). The dashed line represents the 0.3 relative half-width and the numbers represent the required numbers of tests for reaching the 0.3 relative half-width (**e**). **f**, Frequency of the required number of tests for repeated testing experiments for the AV-I model. **g**,**h**, Unweighted crash rate (**g**) and weighted crash rate (**h**) of each crash type in the D2RL-trained testing environment. **i**–**l**, Weighted distributions of the speed difference at the crash moment (**i**), time to collision (**j**), bumper-to-bumper distance (**k**) and post-encroachment time (**l**) of the near-miss events.

the results in the NDE (Fig. 3h), which demonstrates the unbiasedness of our approach within the evaluation precision. Similarly, Fig. 3i–l demonstrates that our approach can also unbiasedly evaluate the AV's safety performance regarding crash severities and near-miss events within the evaluation precision. As near-miss events are critical for the development of AVs, the generated near-miss events without loss of unbiasedness open the door for accelerating the AV training. We leave that for future study.

To further investigate the scalability and generalizability, we conducted the experiments with different numbers of lanes (two and three lanes) and driving distances (400 m, 2 km, 4 km and 25 km) for the AV-I model. Here we studied the 25-km case to demonstrate the effectiveness of our approach over full-length trips, because the average commuter travels approximately 25 km one way in the United States. As shown in Table 1, because of the skipped episodes and

Table 1 | Performance evaluation with different highway simulation environments

		400 m		2 km		4 km		25 km
		Two lanes	Three lanes	Two lanes	Three lanes	Two lanes	Three lanes	Three lanes
NDE	Number of tests	1.9×10^{8}	1.0×10^{8}	4.8×10^{7}	2.5×10^{7}	2.9×10^{7}	9.4×10^{6}	1.7×10^{6}
D2RL	Episodes skipped (%)	95.70	91.73	77.54	79.85	61.42	58.92	8.83
	Steps skipped (%)	99.78	99.70	99.82	99.81	99.79	99.74	99.76
	Number of tests	9.1×10^{4}	4.4×10^{4}	2.4×10^{4}	1.7×10^{4}	1.3×10^{4}	4.5×10^{3}	1.8×10^{3}
	Acceleration ratio	2.1×10^{3}	2.3×10^{3}	2.0×10^{3}	1.5×10^{3}	2.2×10^{3}	2.1×10^{3}	9.4×10^{2}

The numbers of tests of the D2RL approach were the average values of multiple testing experiments, similar to Fig. 3f, and the numbers of tests for the NDE approach were obtained according to the Monte Carlo method[1].

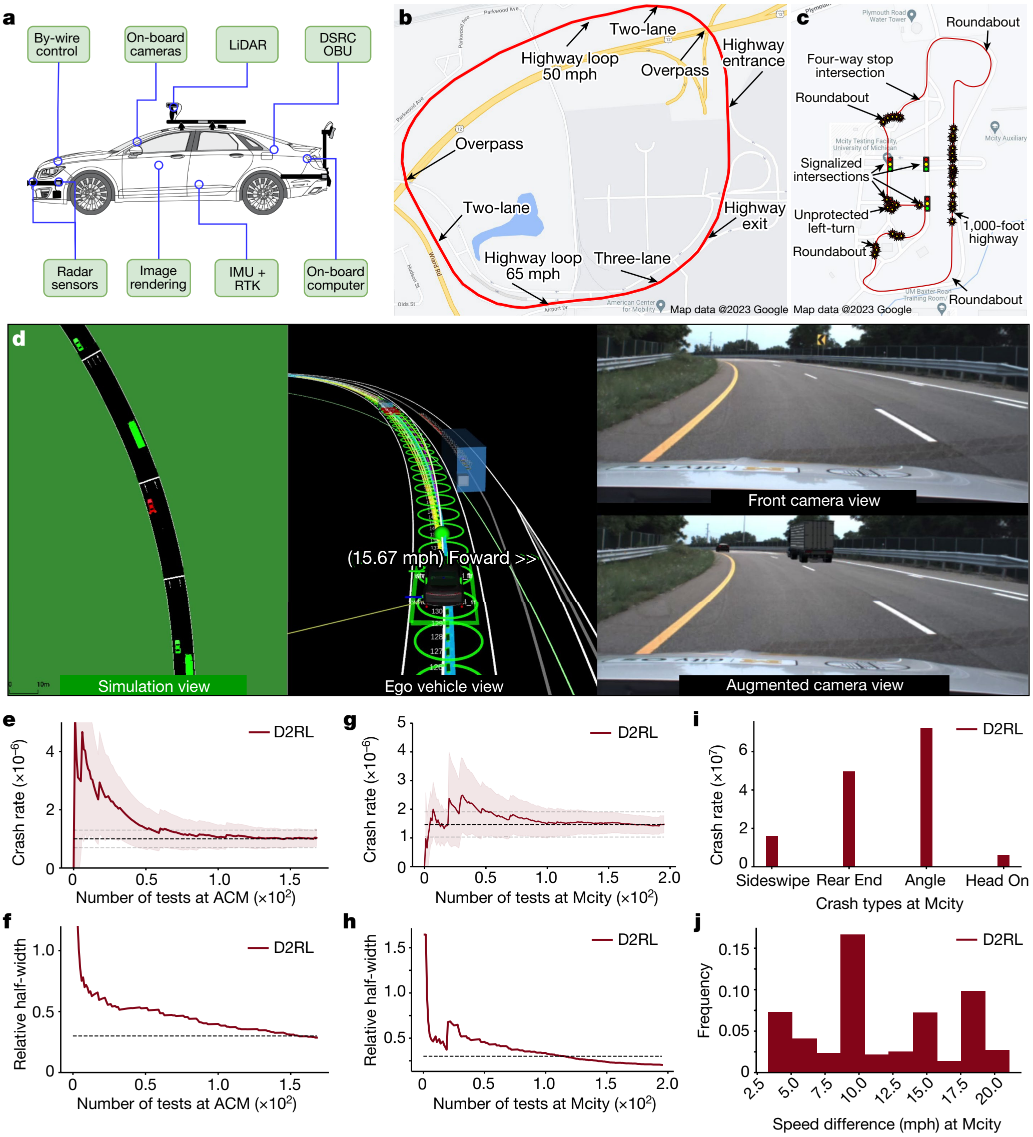

Fig. 4 | Testing experiments for a real-world AV at physical test tracks. **a**, Illustration of the AV under test, equipped with Autoware. IMU, inertial measurement unit; OBU, onboard unit. **b**, Illustration of the ACM highway testing environment. The red line denotes the AV driving route. **c**, Illustration of the Mcity urban testing environment including highways, roundabouts, intersections and so on. The explosion icons denote the locations of crash events that happened during the tests. **d**, Illustration of the real-time visualization of the testing process. Left: the simulation view, where the virtual BVs (green vehicles) are generated and controlled by the intelligent testing environment to interact with the AV (red vehicle). Middle: the real-world AV view visualized by Autoware, where the black vehicle is the AV under test and blue vehicles are augmented BVs. Right: the original image view (top) and augmented image view (bottom) from the AV's front camera. **e**–**h**, Crash rate estimation and the relative half-width of the real AV at the ACM test track (**e**,**f**) and Mcity test track (**g**,**h**) with the augmented-reality testing platform. The black dashed line (**e**,**g**) represents the final estimation of the crash rate, the grey dashed lines (**e**,**g**) represent the 30% relative errors of the crash rate, the grey dashed line (**f**,**h**) represents the 0.3 relative half-width threshold and the shaded areas (**e**,**g**) represents the 90% confidence level. **i**, Crash rates of different crash types of the AV at the Mcity test track. **j**, Distribution of the speed difference at the crash moment for crash severity analysis of the AV at the Mcity test track. Additional explanations regarding the field experiments are provided in Supplementary Videos 3–8.

steps that substantially reduce the training variance, our approach can effectively learn the intelligent testing environment for all the experiments.

Furthermore, to demonstrate the advance of our approach in realistic urban scenarios, we extended our simulation experiments at a real-world four-armed roundabout[32] in Germany with a high traffic volume and complex interactions. Compared with the NDE testing approach that requires about 8.91×10^6 number of tests to reach the 30% relative half-width, our approach only requires 3.76×10^3 number of tests, which is 2.37×10^3 times faster. See Supplementary Video 2 and Supplementary Section 4b for more details.

AV testing in test tracks

Finally, we tested a Lincoln MKZ hybrid equipped with the open-source automated driving system, Autoware[23] (Fig. 4a), driving continuously in the physical multi-lane 4-km highway test track at the ACM (Fig. 4b) and the physical urban test track at Mcity (Fig. 4c). We developed an augmented-reality testing platform[24], which combines the physical test track and a simulation environment, SUMO[25]. As shown in Fig. 1d, by synchronizing the movements of the real AV and virtual BVs, the real AV in the physical test track can interact with the virtual BVs as though it is in a real traffic environment, where the BVs are controlled according to the intelligent testing environment. Figure 4d illustrates the real-time visualization of the testing process. We trained the intelligent testing environment in the digital twins of the ACM highway section and the Mcity urban section using similar training settings to the simulation studies (see Methods for details). As shown in Fig. 4e–h, the crash rate estimations in both the ACM and Mcity converge and reach the 30% relative half-width after about 156 tests at the ACM and 117 tests at Mcity, which are on the order of 10^5 times faster than those (2.5×10^7 at the ACM and 2.1×10^7 at Mcity) of the NDE testing approach. We also evaluated the AV's safety performance for different crash types and severities (Fig. 4i,j).

Discussion

Our results present evidence of using D2RL techniques to validate the safety performance of AVs regarding their behavioural competency[33]. D2RL can accelerate the testing process and can be used for both simulation testing and test-track methods. It can substantially enhance existing testing approaches (falsification methods, scenario-based methods and NDE methods) to overcome their limitations in real-world applications. D2RL also opens the door for leveraging AI techniques to validate machine intelligence of other safety-critical autonomous systems, such as medical robots and aerospace systems.

Ideally, the testing environment should consider all operating conditions of AVs and their associated rare events. For example, a six-layer model[34] has been developed to structure the parameters of scenarios, including road geometry, road furniture and rules, temporal modifications and events, moving objects, environmental conditions, and digital information. In this study, we mainly focus on two layers: moving objects and road geometry, that is, multiple surrounding vehicles undertaking manoeuvres on roads of varying geometry, which are critical for the testing environment. Our approach could be extended to include parameters from other layers, such as weather conditions, by collecting large-scale naturalistic data and utilizing domain knowledge of those fields.

We note that increasing attention has also been paid to formal methods to address the challenges raised by AI systems (see refs. [35,36] and references therein). Formal methods provide a mathematical framework for rigorous system specification, design and verification[37], which are critical for trustworthy AI. However, as discussed in ref. [36], multiple major challenges need to be addressed to fully realize their full potential. D2RL can potentially be integrated with formal methods. For example, reachability-based methods[38] could be incorporated into the calculation of criticality measure to identify the critical states, particularly for generic safety-critical autonomous systems. How to further integrate D2RL with formal methods deserves further investigation.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41586-023-05732-2.

1. Kalra, N. & Paddock, S. M. Driving to safety: how many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transp. Res. A* **94**, 182–193 (2016).
2. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
3. 10 million self-driving cars will be on the road by 2020. *Insider* https://www.businessinsider.com/report-10-million-self-driving-cars-will-be-on-the-road-by-2020-2015-5-6 (2016).
4. Nissan promises self-driving cars by 2020. *Wired* https://www.wired.com/2013/08/nissan-autonomous-drive/ (2014).
5. Tesla's self-driving vehicles are not far off. *Insider* https://www.businessinsider.com/elon-musk-on-teslas-autonomous-cars-2015-9 (2015).
6. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles* (Society of Automotive Engineers, 2021); https://www.sae.org/standards/content/j3016_202104/.
7. *2021 Disengagement Reports* (California Department of Motor Vehicles, 2022); https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/disengagement-reports/.
8. Paz, D., Lai, P. J., Chan, N., Jiang, Y. & Christensen, H. I. Autonomous vehicle benchmarking using unbiased metrics. In *IEEE International Conference on Intelligent Robots and Systems* 6223–6228 (IEEE, 2020).
9. Favarò, F., Eurich, S. & Nader, N. Autonomous vehicles' disengagements: trends, triggers, and regulatory limitations. *Accid. Anal. Prev.* **110**, 136–148 (2018).
10. Riedmaier, S., Ponn, T., Ludwig, D., Schick, B. & Diermeyer, F. Survey on scenario-based safety assessment of automated vehicles. *IEEE Access* **8**, 87456–87477 (2020).
11. Nalic, D. et al. Scenario based testing of automated driving systems: a literature survey. In *Proc. of the FISITA Web Congress* 1–10 (Fisita, 2020).
12. Feng, S., Feng, Y., Yu, C., Zhang, Y. & Liu, H. X. Testing scenario library generation for connected and automated vehicles, part I: methodology. *IEEE Trans. Intell. Transp. Syst.* **22**, 1573–1582 (2020).
13. Feng, S. et al. Testing scenario library generation for connected and automated vehicles, part II: case studies. *IEEE Trans. Intell. Transp. Syst.* **22**, 5635–5647 (2020).
14. Feng, S., Yan, X., Sun, H., Feng, Y. & Liu, H. X. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nat. Commun.* **12**, 748 (2021).
15. Sinha, A., O'Kelly, M., Tedrake, R. & Duchi, J. C. Neural bridge sampling for evaluating safety-critical autonomous systems. *Adv. Neural Inf. Process. Syst.* **33**, 6402–6416 (2020).
16. Li, L. et al. Parallel testing of vehicle intelligence via virtual-real interaction. *Sci. Robot.* **4**, eaaw4106 (2019).
17. Zhao, D. et al. Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques. *IEEE Trans. Intell. Transp. Syst.* **18**, 595–607 (2016).
18. Donoho, D. L. High-dimensional data analysis: the curses and blessings of dimensionality. *AMS Math Challenges Lecture* **1**, 32 (2000).
19. Hinton, G. E. & Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science* **313**, 504–507 (2006).
20. Silver, D. et al. Mastering the game of go without human knowledge. *Nature* **550**, 354–359 (2017).
21. Mirhoseini, A. et al. A graph placement methodology for fast chip design. *Nature* **594**, 207–212 (2021).
22. Cummings, M. L. Rethinking the maturity of artificial intelligence in safety-critical settings. *AI Mag.* **42**, 6–15 (2021).
23. Kato, S. et al. Autoware on board: enabling autonomous vehicles with embedded systems. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems* 287–296 (IEEE, 2018).
24. Feng, S. et al. Safety assessment of highly automated driving systems in test tracks: a new framework. *Accid. Anal. Prev.* **144**, 105664 (2020).
25. Lopez, P. et al. Microscopic traffic simulation using SUMO. In *International Conference on Intelligent Transportation Systems* 2575–2582 (IEEE, 2018).
26. Arun, A., Haque, M. M., Bhaskar, A., Washington, S. & Sayed, T. A systematic mapping review of surrogate safety assessment using traffic conflict techniques. *Accid. Anal. Prev.* **153**, 106016 (2021).
27. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* (MIT Press, 2018).
28. Koren, M., Alsaif, S., Lee, R. & Kochenderfer, M. J. Adaptive stress testing for autonomous vehicles. In *IEEE Intelligent Vehicles Symposium (IV)* 1–7 (IEEE, 2018).
29. Sun, H., Feng, S., Yan, X. & Liu, H. X. Corner case generation and analysis for safety assessment of autonomous vehicles. *Transport. Res. Rec.* **2675**, 587–600 (2021).
30. Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal policy optimization algorithms. Preprint at https://arxiv.org/abs/1707.06347 (2017).

31. Owen, A. B. Monte Carlo theory, methods and examples. *Art Owen* https://artowen.su.domains/mc/ (2013).
32. Krajewski, R., Moers, T., Bock, J., Vater, L. & Eckstein, L. September. The round dataset: a drone dataset of road user trajectories at roundabouts in Germany. *In 2020 IEEE 23rd International Conference on Intelligent Transportation Systems* 1–6 (IEEE, 2020).
33. Nowakowski, C., Shladover, S. E., Chan, C. Y. & Tan, H. S. Development of California regulations to govern testing and operation of automated driving systems. *Transport. Res. Rec.* **2489**, 137–144 (2015).
34. Sauerbier, J., Bock, J., Weber, H. & Eckstein, L. Definition of scenarios for safety validation of automated driving functions. *ATZ Worldwide* **121**, 42–45 (2019).
35. Pek, C., Manzinger, S., Koschi, M. & Althoff, M. Using online verification to prevent autonomous vehicles from causing accidents. *Nat. Mach. Intell.* **2**, 518–528 (2020).
36. Seshia, S. A., Sadigh, D. & Sastry, S. S. Toward verified artificial intelligence. *Commun. ACM* **65**, 46–55 (2022).
37. Wing, J. M. A specifier's introduction to formal methods. *IEEE Comput.* **23**, 8–24 (1990).
38. Li, A., Sun, L., Zhan, W., Tomizuka, M. & Chen, M. Prediction-based reachability for collision avoidance in autonomous driving. In *2021 IEEE International Conference on Robotics and Automation* 7908–7914 (IEEE, 2021).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Methods

Description of the AV safety validation problem

This section describes the problem formulation of AV safety performance evaluation. Denote the variables of the driving environment as $\mathbf{x} = [\mathbf{s}(0), \mathbf{u}(0), \mathbf{u}(1), \cdots, \mathbf{u}(T)]$, where $\mathbf{s}(k)$ denotes the states (position and speed) of the AV and BVs at the kth time step, $\boldsymbol{u}(\mathrm{k})$ denotes the manoeuvres of BVs at the kth time step and T denotes the total time steps of this testing episode. With Markovian assumptions of the BVs' manoeuvres, the probability of each testing episode in the NDE can be calculated as $P(\mathbf{x}) = P(\mathbf{s}(0)) \times \prod_{k=0}^{T} P(\mathbf{u}(k)|\mathbf{s}(k))$, and then the AV crash rate can be measured by the Monte Carlo method[31] as

$$P(A) = \mathbb{E}_{\mathbf{x}\sim P(\mathbf{x})}[P(A|\mathbf{x})] \approx \frac{1}{n}\sum_{i=1}^{n} P(A|\mathbf{x}_i), \mathbf{x}_i \sim P(\mathbf{x}), \quad (2)$$

where A denotes the crash event, n denotes the total number of testing episodes, $i = 1, \ldots, n$ denotes the ith testing episode, and $\mathbf{x}_i \sim P(\mathbf{x})$ indicates that the variables are distributed as $P(\mathbf{x})$. Here a crash is defined as a contact that the subject vehicle (for example, AV) has with an object, either moving or fixed, at any speed resulting in fatality, injury or property damage[39]. As A is a rare event, obtaining a statistically reliable estimation requires a large number of tests (n), which leads to the severe inefficiency issue of the NDE testing approach, as pointed out in ref. [1].

To address this inefficiency issue, the key is to generate an intelligent driving environment, where BVs can be controlled purposely to test the AV unbiasedly and efficiently. In essence, testing an AV in the intelligent driving environment is to estimate $P(A)$ in equation (2) by the importance sampling method[31] as

$$P(A) = \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}[P(A|\mathbf{x}) \times W_q(\mathbf{x})] \approx \frac{1}{n}\sum_{i=1}^{n} P(A|\mathbf{x}_i) \times W_q(\mathbf{x}_i), \mathbf{x}_i \sim q(\mathbf{x}), \quad (3)$$

where $q(\mathbf{x})$ denotes the underlying distribution of BVs' manoeuvres in the intelligent testing environment, and $W_q(\mathbf{x})$ is the likelihood of each testing episode as

$$W_q(\mathbf{x}) = \frac{P(\mathbf{x})}{q(\mathbf{x})} = \prod_{k=0}^{T}\left[\frac{P(\mathbf{u}(k)|\mathbf{s}(k))}{q(\mathbf{u}(k)|\mathbf{s}(k))}\right]. \quad (4)$$

According to the importance sampling theory[31], the unbiasedness of the estimation in equation (3) can be guaranteed if $q(\mathbf{x}) > 0$ for any $\mathbf{x}$ that $P(A|\mathbf{x})P(\mathbf{x}) > 0$. To optimize the estimation efficiency, the importance function $q(\mathbf{x})$ needs to minimize the estimation variance

$$\sigma_q^2 = \mathbb{E}_q(P^2(A|\mathbf{x}) \times W_q^2(\mathbf{x})) - P^2(A). \quad (5)$$

Therefore, the generation of the intelligent testing environment is formulated as a sequential MDP problem of the BVs' manoeuvres (that is, determine $q(\mathbf{u}(k)|\mathbf{s}(k))$ to minimize the estimation variance σ_q^2 in equation (5). However, how to solve such a sequential MDP problem associated with rare events and high-dimensional variables remains a highly challenging problem, and most existing importance sampling-based methods suffer from the curse of dimensionality[40], where the estimation variance would increase exponentially with the dimensionality. In our previous study[14], we found that the curse of dimensionality issue could be addressed theoretically by sparse adversarial control to the naturalistic distribution. However, only a model-based method with handcrafted heuristics was utilized for conducting the sparse adversarial control, which suffers from substantial spatiotemporal limitations, and how to leverage AI techniques to train the BVs for truly learning the testing intelligence remains unsolved, which is the focus of this paper. More details of related work can be found in Supplementary Section 1.

Formulation as a deep-reinforcement-learning problem

This section describes how to generate the intelligent testing environment as a DRL problem. As mentioned above, the goal is to minimize the estimation variance in equation (5) by training a policy π modelled by a neural network θ that can control BVs' manoeuvres with the underlying distribution $q_\pi(\mathbf{u}|\mathbf{s})$. To keep the notation simple, we leave it implicit in all cases that π is a function of θ. An MDP usually consists of four key elements: state, action, state transition and reward. In this study, states encode information (position and speed) about the AV and surrounding BVs, actions include 31 discrete longitudinal accelerations ([−4, 2] with 0.2 m s^{-2} discrete resolution), left lane change and right lane change, and state transitions define the probability distribution over next states that are also dependent on the AV manoeuvre. Here we assumed that a lane-change manoeuvre of BVs would be initiated from its current position and completed in one second if a lane-change action was decided. Our framework is also applicable to more realistic and complex action settings.

For the corner-case-generation case study, we studied a three-lane highway driving environment, where eight critical BVs (that is, principal other vehicles or POVs) are controlled to interact with the AV for a certain distance (400 m) and each BV has the 33 discrete actions at every 0.1 s. For the intelligent-testing-environment generation case study, to keep the runtime of the DRL small, we simplified the output of the neural network as the adversarial manoeuvre probability ($\varepsilon_\pi \in (0, 1)$) of the most critical POV (Principal Other Vehicle), whereas POV's other manoeuvres are normalized by $1 - \varepsilon_\pi$ according to the naturalistic distribution and other BVs' manoeuvres keep following the naturalistic distribution. The adversarial manoeuvre and POV are determined by the criticality measure. We note that the generalization of this work to multiple POVs is straightforward.

The reward function design is critical for the DRL problem[41]. As the goal of the intelligent testing environment is to minimize the estimation variance in equation (5), we derived the objective function of the DRL problem as

$$\min_q \sigma_q^2 = \max_\pi\{-\mathbb{E}_{q_{\pi_\mathrm{b}}}(\mathbb{I}_A(\mathbf{x}) \times W_{q_\pi}(\mathbf{x}) \times W_{q_{\pi_\mathrm{b}}}(\mathbf{x}))\}, \quad (6)$$

where $\mathbb{I}_A$ is the indicator function of the crash event and π_b denotes the behaviour policy of the DRL. During the training process, the training data are collected by the behaviour policy, which is a Monte Carlo estimation of the expectation in equation (6), so we can obtain the reward function as

$$r(\mathbf{x}) = -\mathbb{I}_A(\mathbf{x}) \times W_{q_\pi}(\mathbf{x}) \times W_{q_{\pi_\mathrm{b}}}(\mathbf{x}), \quad (7)$$

which is theoretically consistent with the objective function. As it is mainly based on the importance sampling theory, the reward function is also applicable to other rare-event estimation problems with high-dimensional variables. To limit the scale of the error derivatives[42], we rescaled and clipped the function, resulting in the reward function that belongs to [−100, 100], where the scaling constants could be automatically determined during the learning process.

With the state, action, state transition and reward function, the intelligent-testing-environment generation problem becomes a DRL problem. However, as the gradient estimation of neural networks would suffer from the large variance due to the rareness of informative data, applying learning-based techniques for safety-critical systems is highly challenging because of the curse of rarity. It is hard—or even empirically infeasible—to learn an effective policy if directly applying DRL approaches.

Dense deep reinforcement learning

To address this challenge, we propose the D2RL approach in this paper. Specifically, according to the policy gradient theorem[27], the policy gradient of the objective function for DRL approaches can be estimated as

$$\nabla \hat{J}(\theta) = \hat{q}_{\pi}(S_t, A_t)\frac{\nabla \pi(A_t|S_t, \theta)}{\pi(A_t|S_t, \theta)}, \tag{8}$$

where θ denotes the parameters of the policy, $q_{\pi}(S_t, A_t)$ denotes the state–action value, S_t and A_t are samples of the state and action under the policy at time t, $\hat{q}_{\pi}(S_t, A_t)$ is an unbiased estimation of $q_{\pi}(S_t, A_t)$, that is, $\mathbb{E}_{\pi}[\hat{q}_{\pi}(S_t, A_t)] = q_{\pi}(S_t, A_t)$. Differently, for the D2RL approach, we propose to estimate the policy gradient as

$$\nabla_{\mathrm{dense}} \hat{J}(\theta) = \hat{q}_{\pi}(S_t, A_t)\frac{\nabla \pi(A_t|S_t, \theta)}{\pi(A_t|S_t, \theta)}\mathbb{I}_{S_t \in \mathbb{S}_{\mathrm{c}}}, \tag{9}$$

where $\mathbb{S}_{\mathrm{c}}$ denotes the set of critical states and $\mathbb{I}_{S_t \in \mathbb{S}_{\mathrm{c}}}$ denotes the indicator function.

Here, a state is defined as an uncritical state if $\upsilon_{\pi}(s) = q_{\pi}(s, a), \forall a$, where s denotes the state, a denotes the action, $\upsilon_{\pi}(s) \overset{\mathrm{def}}{=} \mathbb{E}_{\pi}(q_{\pi}(s, a))$ denotes the state value, so the set of critical states can be defined as $\mathbb{S}_{\mathrm{c}} \overset{\mathrm{def}}{=} \{s|\upsilon_{\pi}(s) \neq q_{\pi}(s, a), \exists a\}$. It indicates that a state is defined as uncritical if any action (for example, BVs' manoeuvres) from the current state will not affect the expected value of the state (for example, AV's crash probability within a specific time horizon from the current state). We note that this definition is primarily for the theoretical analysis to be clean and is not strictly required to run the algorithm in practice. For example, a state can be practically identified as uncritical if the current action will not substantially affect the expected value of the state. For specific applications, the critical states can be approximately identified based on domain-specific models or physics. For example, the criticality measure[12,13], which is an outer approximation of the AV crash rate within a specific time horizon (for example, one second), is utilized in this study to demonstrate the approach for the AV testing problem. We note that many other safety metrics[26] could also be applicable, such as the model predictive instantaneous safety metric[43] developed by the National Highway Traffic Administration in the United States and the criticality metric[44] developed by the PEGASUS project in Germany, as long as the identified set of states covers the critical states. More theoretical analysis for a more general sense can be found in Supplementary Section 2a.

Then, we have the following theorem, and the proof can be found in Supplementary Information.

Theorem 1

The policy gradient estimator of D2RL has the following properties:

(1) $\mathbb{E}_{\pi}[\nabla_{\mathrm{dense}} \hat{J}(\theta)] = \mathbb{E}_{\pi}[\nabla \hat{J}(\theta)]$;

(2) $\mathrm{Var}_{\pi}[\nabla_{\mathrm{dense}} \hat{J}(\theta)] \leq \mathrm{Var}_{\pi}[\nabla \hat{J}(\theta)]$; and

(3) $\mathrm{Var}_{\pi}[\nabla_{\mathrm{dense}} \hat{J}(\theta)] \leq \rho_{\pi}\mathrm{Var}_{\pi}[\nabla \hat{J}(\theta)]$, with the assumption

$$\mathbb{E}_{\pi}[\sigma^2_{\pi}(S_t, A_t)\mathbb{I}_{S_t \in \mathbb{S}_{\mathrm{c}}}] = \mathbb{E}_{\pi}[\sigma^2_{\pi}(S_t, A_t)]\mathbb{E}_{\pi}[\mathbb{I}_{S_t \in \mathbb{S}_{\mathrm{c}}}], \tag{10}$$

where $\rho_{\pi} \overset{\mathrm{def}}{=} \mathbb{E}_{\pi}(\mathbb{I}_{S_t \in \mathbb{S}_{\mathrm{c}}}) \in [0, 1]$ is the proportion of critical states in all states under the policy π (for example, $1 - \rho_{\pi}$ denotes the proportion of steps skipped in Fig. 2b and Table 1), and $\sigma^2_{\pi}(S_t, A_t) = \left(\hat{q}_{\pi}(S_t, A_t)\frac{\nabla \pi(A_t|S_t, \theta)}{\pi(A_t|S_t, \theta)}\right)^2$.

Theorem 1 proves that the D2RL approach has an unbiased and efficient estimation of the policy gradient compared with the DRL approach. To quantify the variance reduction of dense learning, we introduce the assumption in equation (10), which assumes that $\sigma^2_{\pi}(S_t, A_t)$ is independent on the indicator function $\mathbb{I}_{S_t \in \mathbb{S}_{\mathrm{c}}}$. As both the policy and the state–action values are randomly initialized, the values of $\sigma^2_{\pi}(S_t, A_t)$ are quite similar for all different states, so the assumption is valid at the early stage of the training process. Such variance reduction will enable the D2RL approach to optimize the neural network, whereas the DRL approach would be stuck at the beginning of the training process.

We then consider the influence of dense learning on estimating $\hat{q}_{\pi}(S_t, A_t)$ with bootstrapping, which can guide the information propagation in the state–action space. For example, the fixed-length advantage estimator ($\hat{A}_t$) is commonly used for the PPO algorithm[30] as

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{L-t+1}\delta_{L-1}, \tag{11}$$

where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$, $V(s_t)$ is the state–value function, γ denotes the discount rate, and L denotes the fixed length. For safety-critical applications, the immediate reward is usually zero (that is, $r_t = 0$), and most state–value functions are determined by initial random values without any valuable information because of the rarity of events. Bootstrapping with such noisy state–value functions will not be effective in the learning process. By editing the Markov chain, only the critical states will be considered. Then, the advantage estimator will be essentially modified as

$$\overline{A}_t = \delta_{z(t,0)} + (\gamma\lambda)\delta_{z(t,1)} + \cdots + (\gamma\lambda)^{L-t+1}\delta_{z(t,L-1)}, \tag{12}$$

where $\delta_{z(t,j)} = r_{z(t,j)} + \gamma V(s_{z(t,j+1)}) - V(s_{z(t,j)})$, j is a natural number, and z is a function that $z(t, 0) = t$, $z(t,j) = \min_i\{s_i \in \mathbb{S}_{\mathrm{c}}|i > z(t,j-1)\}, j > 0$, and i is a natural number. In essence, it is a state-dependent temporal-difference learning, where only the values of critical states are utilized for bootstrapping. As the critical states have much higher probabilities leading to safety-critical events, the reward information can be propagated to these critical state values more easily. Utilizing the values of these critical states, the bootstrapping can guide the information from the safety-critical events to the state–action space more efficiently. This mechanism can help avoid the interference of the large number of noisy data and focus the policy on learning the sparse but valuable information. Because of the abovementioned variance reductions regarding the policy gradient estimation and bootstrapping, the D2RL approach substantially improves the learning effectiveness compared with the DRL approach, enabling the neural network to learn from the safety-critical events.

Densifying the information is a natural way to overcome the challenges caused by the rarity of events. In the field of deep neural networks, connecting different layers of neural networks more densely has been demonstrated to produce better training efficiency and efficacy, that is, DenseNet[45]. Instead of connecting layers of neural networks, our approach densifies the information by connecting states more densely with safety-critical states, besides the natural connections provided by the state transitions. As safety-critical states have more connections with rare events, they contain more valuable information with less variance. By densifying the connections between safety-critical states with other states, we can better propagate the valuable information to the entire state space, which can substantially facilitate the learning process. This study proposed and demonstrated one specific realization of the dense-learning approach by approximately identifying uncritical states and connecting the remaining states directly. This can be further improved by more flexible and dense connections among safety-critical states and uncritical states. The connections can even be added in the form of curriculum learning[46], which can guide the information propagation gradually. The measures for identifying critical states can also be further improved by involving more advanced modelling techniques.

Off-policy learning mechanism

We justify the off-policy learning mechanism in this section. The goal of the behaviour policy π_{b} is to collect training data for improving the target policy π that can maximize the objective function in equation (6). To achieve this goal, it is critical to estimate the objective function accurately using the reward function in equation (7), which determines the calculation of the policy gradient. However, only episodes with crashes have non-zero rewards, so the objective function estimation suffers from a large variance, because of the rarity of crashes. Without an accurate estimation of the objective function, the training could be misled. According to the importance sampling theory, we have the following theorem, and the proof can be found in Supplementary Information.

Theorem 2

The optimal behaviour policy π_{b}^* that can minimize the estimation variance of the objective function has the following property:

$$q_{\pi_{\mathrm{b}}^*}(\mathbf{x}) \propto \frac{q^2_{\pi^*}(\mathbf{x})}{q_\pi(\mathbf{x})}, \tag{13}$$

where $q_{\pi^*}(\mathbf{x})$ denotes the optimal importance sampling function that is unchanged during the training process, and the symbol ∝ means 'proportional to'.

Theorem 2 finds that the optimal behaviour policy is nearly inversely proportional to the target policy, particularly at the beginning of the training process when q_π is far from q_{π^*}. If using on-policy learning mechanisms ($q_{\pi_{\mathrm{b}}} = q_\pi$), the behaviour policy would be far from optimality, which could mislead the training process and eventually cause the underestimation issues. For example, if a target policy misses an action that could lead to a likely crash, an on-policy learning mechanism will never find this missing crash. More importantly, the on-policy mechanism could mislead the policy for purposely hiding the crashes that are difficult to evaluate, leading to the severe underestimation issue of the safety performance evaluation.

We design an off-policy learning mechanism to address this issue, where a generic behaviour policy is designed and kept unchanged during the training process. Specifically, we determined a constant probability of the adversarial manoeuvre of the POV (that is, $\varepsilon_{\pi_{\mathrm{b}}} = 0.01$) and conducted other manoeuvres with the total probability of 0.99 that were normalized according to the naturalistic distribution. This policy explores the state–action space using the naturalistic distribution most of the time and exploits the information of the model-based criticality measure that helps identify the POV and adversarial manoeuvre. We note that although the optimal behaviour policy needs to be adaptively determined based on the target policy, as indicated in Theorem 2, an off-policy learning mechanism can provide a sufficiently good foundation for effective learning in this study. The behaviour policy is also not sensitive to the constant of ε_{π_b}, and generally, a small value (for example, 0.1, 0.05, 0.01 and so on) that balances the exploration and exploitation would be effective in this study. Further improvement can be investigated in the future.

Simulation settings

NDE simulator. To simulate the NDE, we developed a simulation platform based on an open-source traffic simulator SUMO. The scheme of the platform can be found in Supplementary Information. We utilized both the C++ and TRACI interfaces to refine the SUMO simulator so that high-fidelity driving environments can be integrated. Specifically, we rewrote and recompiled the C++ codes of SUMO to integrate the high-fidelity driving environments, including car-following and lane-changing behaviour models. Then, we utilized the TRACI interface to implement the intelligent testing environment, where at selected moments, selected vehicles would execute specific adversarial manoeuvres with a learned probability, following the policy obtained by the D2RL approach. We also synchronized the modified SUMO with the physical test tracks related to the information of BVs, AVs, traffic signals, high-definition maps and so on, through the TRACI interface. To provide a training environment for intelligent testing environments, we constructed a multi-lane highway driving environment and an urban driving environment, where all vehicles were controlled at 100-ms intervals.

Driving behaviour models in the NDE simulator. The default driving behaviour models of SUMO, which are simple and deterministic, cannot be utilized for safety testing and training of AVs because they are designed to be crash-free models. To address this issue, in this study, we constructed NDE models[47] to provide naturalistic behaviours of BVs according to the large-scale naturalistic driving datasets (NDDs) from the Safety Pilot Model Deployment programme[48] and the Integrated Vehicle-Based Safety System programme[49] at the University of Michigan, Ann Arbor. At each step of simulation, the NDE models can provide distributions of each BV's manoeuvres, which are consistent with the NDD. Then, by sampling manoeuvres from the distributions, a testing environment that can evaluate the real-world safety performance can be generated. For the field testing at ACM and Mcity, although the intelligent testing environment can accelerate the AV testing from about 10^7 loops of testing to only around 10^4 loops (Table 1), this still represents a substantial level of effort for an academic research group. To demonstrate our approach in a more efficient way, we simplified the NDE models to demonstrate our method more conveniently. Specifically, we modified the Intelligent Driving Model (IDM)[50] and the Minimizing Overall Braking Induced by Lane change (MOBIL) model[51] as stochastic models to construct the simplified NDE models. More details of the NDE models can be found in Supplementary Information.

D2RL architecture, implementation and training. The D2RL algorithm can be easily plugged into existing DRL algorithms by defining a specific environment with the dense-learning approach. Specifically, for existing DRL algorithms, the environment receives the decision from the DRL agent, executes the decision, and then collects observations and rewards at each time step, whereas for the D2RL algorithm, the environment collects only the observations and rewards for the critical states, as illustrated in Supplementary Section 3e. In this way, we can quickly implement the D2RL algorithm utilizing existing DRL platforms. In this study, we utilized the PPO algorithm implemented at the RLLib 1.2.0 platform[52], which was parallelly trained on 500 central-processing-unit cores and 3,500-GB memory high-performance computation cluster at the University of Michigan, Ann Arbor. We designed a three-layer fully connected neural network with 256 neurons in each layer and chose the 10^{-4} learning rate and 1.0 discount factor besides the default parameters. Each central processing unit collected 120 time steps of training data for all experiment settings in each training iteration, so a total of 60,000 time steps were collected in each training iteration. For the corner-case generation, the neural network's output is the actions of the closest 8 BVs, where each BV has the 33 discrete actions space: left lane change, 31 discrete longitudinal accelerations ([−4, 2] with 0.2 m s^{-2} discrete resolution) and right lane change. For the intelligent-testing-environment generation, the neural network's output is the adversarial manoeuvre probability (ε_π) of the POV, where the action space is $\varepsilon_\pi \in [0.001, 0.999]$. To further improve the data efficiency during the training process, we used the collected data with a resampling mechanism to train the neural network for multiple steps.

Field test settings

Augmented-reality testing platform. We implemented the augmented-reality testing platform at the ACM, one of the world's premier test tracks for AVs located in Ypsilanti, Michigan, and the Mcity test track, which is the world's first purpose-built test track for AV testing. In this study, we utilized the 4-km highway loop featuring two and three lanes and both exit and entrance ramps to create various merging opportunities, as well as the Mcity urban driving environment, including various types of highway, roundabout, urban streets and so on, as shown in Supplementary Section 3f. We constructed digital twins of the ACM and Mcity based on the NDE simulator and available high-definition maps. To synchronize the information between the simulation and physical test track, we utilized thededicated short-range communications (DSRC) roadside units that were installed in the test tracks. These DSRC-based devices can communicate with AVs via 802.11p and SAE J2735 protocols through the immediate-forward-messaging and forwarding functions. Specifically, we utilized the immediate-forward-messaging function to broadcastproxy basic safety messages (BSMs) containing virtual BVs' identifier, latitude, longitude, altitude and so on, to the physical AV, and the forwarding function to forward incoming BSMs of the AV to the digital twins. After receiving the BSMs of the AV, we synchronized the AV states in the simulation world, where BVs were controlled by the intelligent testing environment. More details of the platform can be found in ref. [24]. We implemented the system with an average 33-ms communication delay, which is acceptable for AV testing and can be further improved with advanced wireless communication techniques.

Augmented image rendering. We use augmented-reality techniques to render and blend virtual objects (for example, vehicles) onto the camera view of the ego vehicle. Given a background three-dimensional model with its 6 degrees of freedom pose/location in the world coordinate, we perform a two-stage transformation to project the model to the onboard camera image: (1) from the world coordinate to the ego-vehicle coordinate, and (2) from the ego-vehicle coordinate to the onboard camera coordinate. In the first transformation, the ego vehicle pose and location are obtained from the real-time signal of the onboard high-precision real-time kinematic positioning (RTK). In the second transformation, the projection is based on the pre-calibrated camera intrinsic and extrinsic. We also perform relighting on the rendered layer to harmonize the visual quality of the blending result. The augmented view is generated based on a linear blending with the rendered foreground layer, the camera's background layer and the rendered alpha matte. On top of the blending result, a weather-control layer is further added to simulate different weather conditions, for example, rain, snow and fog. We implemented the augmented rendering based on pyrender[53]. An additional validation of the augmented image rendering can be found in Supplementary Section 4f.

AV under test. As the AV under test, we used a retrofitted Lincoln MKZ from the Mcity Test Facility at the University of Michigan, Ann Arbor. The vehicle was equipped with multiple sensors, computing resources (two Nexcom Lumina) and with drive-by-wire capabilities provided by Dataspeed Inc. Specifically, the sensors include a PointGrey camera, a Velodyne 32-channel LiDAR, Delphi radars, OTXS RT3003 RTK GPS, Xsens MTi GPS/inertial measurement unit and so on. We implemented the vehicle with a Robot Operating System-based open-source software, Autoware.AI[23], which provides full-stack software for the highly automated driving functions, including localization, perception, planning, control and so on. We then integrated the AV with the augmented-reality testing platform to evaluate the AV's safety performance. An illustration of the system framework can be found in Supplementary Information. Specifically, we modified the AV localization component to utilize the high-definition map and high-accuracy RTK for obtaining the current pose and velocity. The surrounding vehicles' BSMs were directly obtained from the simulation through wireless communications. To generate the AV's future trajectory, we applied the OpenPlanner 1.13[54] as the decision module, an advanced planning algorithm including global and local path planning. We applied the pure pursuit algorithm to convert the planned trajectory into the velocity and yaw rate and then used a proportional–integral–derivative controller provided by Dataspeed Inc. to further convert them into the vehicle by-wire control commands, that is, steering angle, throttle and brake percentages.

Data availability

The raw datasets that we used for modelling the naturalistic driving environment come from the Safety Pilot Model Deployment (SPMD) programme[48] and the Integrated Vehicle-Based Safety System (IVBSS)[49] at the University of Michigan, Ann Arbor. The ShapeNet Dataset that includes the three-dimensional model assets for the image augmented-reality module can be found at https://github.com/mmatl/pyrender. The police crash reports used in Supplementary Video 7 are available at https://www.michigantrafficcrashfacts.org/. The processed data for constructing NDE models and the intelligent testing environment and the experiment results that support the findings of this study are available at https://github.com/michigan-traffic-lab/Dense-Deep-Reinforcement-Learning. Source data are provided with this paper.

Code availability

The simulation software SUMO, the automated driving system Autoware and the RLLib platform with the implemented PPO algorithm are publicly available, as described in the text and the relevant references[23,25,52]. The source codes for the naturalistic driving environment simulator, the driving behaviour models in the simulator, the D2RL-based intelligent testing environment and the simulation set-ups are available at https://github.com/michigan-traffic-lab/Dense-Deep-Reinforcement-Learning.

39. Automated Vehicle Safety Consortium *AVSC Best Practice for Metrics and Methods for Assessing Safety Performance of Automated Driving Systems (ADS)* (SAE Industry Technologies Consortia, 2021).
40. Au, S. K. & Beck, J. L. Important sampling in high dimensions. *Struct. Saf.* **25**, 139–163 (2003).
41. Silver, D., Singh, S., Precup, D. & Sutton, R. S. Reward is enough. *Artif. Intell.* **299**, 1–13 (2021).
42. Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
43. Weng, B., Rao, S. J., Deosthale, E., Schnelle, S. & Barickman, F. Model predictive instantaneous safety metric for evaluation of automated driving systems. In *IEEE Intelligent Vehicles Symposium (IV)* 1899–1906 (IEEE, 2020).
44. Junietz, P., Bonakdar, F., Klamann, B. & Winner, H. Criticality metric for the safety validation of automated driving using model predictive trajectory optimization. In *International Conference on Intelligent Transportation Systems* 60–65 (IEEE, 2018).
45. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition* 4700–4708 (IEEE, 2017).
46. Bengio, Y., Louradour, J., Collobert, R. & Weston, J. Curriculum learning. In *International Conference on Machine Learning* 41–48 (ICML, 2009).
47. Yan, X., Feng, S., Sun, H., & Liu, H. X. Distributionally consistent simulation of naturalistic driving environment for autonomous vehicle testing. Preprint at https://arxiv.org/abs/2101.02828 (2021).
48. Bezzina, D. & Sayer, J. *Safety Pilot Model Deployment: Test Conductor Team Report* DOT HS 812 171 (National Highway Traffic Safety Administration, 2014).
49. Sayer, J. et al. *Integrated Vehicle-based Safety Systems Field Operational Test: Final Program Report* FHWA-JPO-11-150; UMTRI-2010-36 (Joint Program Office for Intelligent Transportation Systems, 2011).
50. Treiber, M., Hennecke, A. & Helbing, D. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E* **62**, 1805 (2000).
51. Kesting, A., Treiber, M. & Helbing, D. General lane-changing model MOBIL for car-following models. *Transp. Res. Rec.* **1999**, 86–94 (2007).
52. Liang, E. et al. RLlib: abstractions for distributed reinforcement learning. In *International Conference on Machine Learning* 3053–3062 (ICML, 2018).
53. Chang A. X. et al. ShapeNet: an information-rich 3D model repository. Preprint at https://arxiv.org/abs/1512.03012 (2015).
54. Darweesh, H. et al. Open source integrated planner for autonomous navigation in highly dynamic environments. *J. Robot. Mechatron.* **29**, 668–684 (2017).

Acknowledgements This research was partially funded by the US Department of Transportation (USDOT) Region 5 University Transportation Center: Center for Connected and Automated Transportation (CCAT) of the University of Michigan (#69A3551747105) and the National Science Foundation (CMMI #2223517). We thank the American Center for Mobility (ACM) for providing access to their test track. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the official policy or position of the US government or the American Center for Mobility.

Author contributions S.F. and H.X.L. conceived and led the research programme, developed the AI against AI concepts, developed the dense-learning approach, and wrote the paper. S.F. and H.S. developed the algorithms for the intelligent-testing-environment generation and

designed the experiments. H.S. and H.Z. developed the simulation platform, implemented the algorithms, performed the simulation tests and prepared the simulation results. X.Y., H.Z. and S.S. implemented the Autoware system in the autonomous vehicle, performed the field tests and prepared the testing results. Z.Z. developed and performed the augmented image rendering. All authors provided feedback during the manuscript revision and results discussions. H.X.L. approved the submission and accepted responsibility for the overall integrity of the paper.

Competing interests The University of Michigan is in the process of applying for a patent application #63/338,424 covering the dense reinforcement learning, intelligent testing environment generation, and augmented reality testing techniques that lists H.X.L., S.F., H.S., X.Y., H.Z., Z.Z., and S.S. as inventors.

Additional information
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41586-023-05732-2.
Correspondence and requests for materials should be addressed to Henry X. Liu.
Peer review information *Nature* thanks Colin Paterson, Fredrik Warg and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.
Reprints and permissions information is available at http://www.nature.com/reprints.