Shifting Deep Reinforcement Learning Algorithm Toward Training Directly in Transient Real-World Environment: A Case Study in Powertrain Control

Bo Hu and Jiaxi Li

***Abstract*—Deep reinforcement learning (DRL) excels at playing a wide variety of simulated games and allows for a generic learning process that does not consider a specific knowledge of the task. However, due to the fact that a large prohibitively number of interactions with the environment are required and that the initial policy behavior is almost random, such an algorithm cannot be trained directly in a real-world environment while satisfying given safety constraints. In this article, a control framework based on DRL that shifts toward training directly in the transient real-world environment is proposed. This research is working on the assumption that some demonstration knowledge that operates under previous controllers and an abstract of the agent environment dynamics are available. By encoding this prior knowledge into a sophisticated learning architecture, a warm-starting DRL algorithm with a safe exploration guarantee can be anticipated. Taking the boost control problem for a variable geometry turbocharger equipped diesel engine as an example, the proposed algorithm improves the initial performance by 74.6% and the learning efficiency by an order of magnitude in contrast to its vanilla counterpart. Compared with other existing DRL-based powertrain control methods, the proposed algorithm can realize the "model-free" concept in the strict sense, making it attractive for future DRL-based powertrain control algorithms to build on.**

***Index Terms*—Deep reinforcement learning (DRL), demonstration, powertrain control, real-world environment.**

I. Introduction

REINFORCEMENT learning (RL) has long been considered as an approach; animals use it to interact with the surrounding environments to realize optimal behavior based on the reward feedback. When RL was first introduced [1], simplified tabular solutions were developed that operate with little prior knowledge and, thus, can only be applied to simulated problems in which the state and action spaces are small. Recent years have seen a rise in demand for RL agents capable of performing complex actions in continuous environments [2]. Deep reinforcement learning (DRL) combines deep learning technology and approximate-solution-based RL and it is considered as one hot orientation of today's machine learning research [3]. It appears to offer a viable path for solving many decision-making problems that cannot currently be solved by any other approaches. For instance, DRL excels at solving a wide variety of Atari and board games and some of them can achieve superhuman performance, taking AlphaGo and AlphaGo Zero [4] for example. Due to the fact that DRL allows for a generic learning process that does not consider a specific knowledge of the task, it is rapidly gaining attention and has opened up a new window for many industrial control problems [5].

However, most of the DRL algorithms successfully applied to the field of simulated games currently cannot be directly migrated to tasks operated in the real physical environment [6]. This is mainly because most of the proposed DRL algorithms do not employ prior knowledge to kickstart learning, i.e., the best a DRL controller can initially do is to "trial and error" uniformly at random and learn from scratch. Therefore, a large prohibitively number of interactions with the environment are required for a DRL-based controller to reach a desirable level of performance. For example, AlphaGo Zero, the evolved version of AlphaGo, requires 4.9 million self-matches to reach the master level [4]. This may be acceptable for a simulator but severely pose limitations to many industrial tasks, whose controller has to be trained in the real-world environment. Furthermore, safety is a key challenge for many industrial problems and erroneous behavior in safety critical systems may inflict serious consequences in the real

Manuscript received November 10, 2020; revised February 5, 2021; accepted February 26, 2021. Date of publication March 3, 2021; date of current version August 20, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 51905061, in part by the Natural Science Foundation of Chongqing under Grant cstc2019jcyj-msxmX0097, in part by China Postdoctoral Science Foundation under Grant 2020M671842, and in part by the Science of Technology Research Program of Chongqing Municipal Education Commission under Grant KJQN201801124. Paper no. TII-20-5161. *(Bo Hu and Jiaxi Li contributed equally to this work.) (Corresponding author: Bo Hu.)*

Bo Hu is with the Key Laboratory of Advanced Manufacturing Technology for Automobile Parts, Ministry of Education, Chongqing University of Technology, Chongqing 400054, China, and also with Ningbo Yinzhou DLT Technology, Company, Ltd., Ningbo 315000, China (e-mail: b.hu@cqut.edu.cn).

Jiaxi Li is with the Key Laboratory of Advanced Manufacturing Technology for Automobile Parts, Ministry of Education, Chongqing University of Technology, Chongqing 400054, China (e-mail: 11607990404@2016.cqut.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2021.3063489.

Digital Object Identifier 10.1109/TII.2021.3063489

physical world. In this context, a complementary mechanism to monitor and interfere with its operation whenever absolutely needed is required in order to get the utmost out of the DRL algorithm while ensuring safety simultaneously [7].

A. Simulation-to-Real Transfer Technique

Although the nature of DRL makes it attractive for a wide range of potential industrial applications, including vehicle powertrain control [8], energy management of hybrid electric vehicle [9], [10], autonomous vehicle [11], intelligent transportation system [12], assistive robots [13], unmanned aerial vehicle autonomous target search [14], and industrial internet of things [15], most of the current research articles only focus on the "simulation-to-real" transfer learning technique, that is pretraining the control algorithm offline in a high-fidelity simulation environment and after that transferring the control parameters from the simulator to an online real-world controller for fine-tuning (some may skip the fine-tuning phase and use the control parameters of the learned controller in the simulation directly). However, for many industrial problems, simulation model may be unavailable and the internal representation (and/or capability) between the simulated and real agent can be very different. In this context, the difficulty in building complex industrial systems will cause a shift toward training a DRL-based control strategy directly in the real-world environment.

B. Learning From Demonstration Technique

While high-fidelity simulation models are difficult to build, most of the industrial powertrain control problems have the data of the control system operating under previous controllers that behave suboptimally [16]. In general, most of the current DRL research makes use of the demonstration knowledge from the perspective of either exploitation or exploration.

The most intuitive way to exploit the demonstration data is via DRL policy initialization using supervised learning (sometimes termed imitation learning or behavior cloning) with no reward signal that allows the agent to evaluate its behavior. This typically requires a large number of teacher demonstrations from which a policy that reproduces and generalizes the demonstration can be derived. However, the prediction errors made in different states could add up; therefore, a mistake made by the controller can easily put it into a state that is far from the demonstration [17]. In addition, the behavior of demonstrations themselves can be imperfect, limiting the quality of the policies derived from them [18]. The other initialization-oriented exploitation approach is by means of DRL pretraining, assuming that the demonstration data were derived from real interactions with the environment [19]. In order to realize a vanilla DRL learning process, the reward function that defines the task problems is incorporated into the demonstration experience. By doing this, the networks of a DRL policy can be updated offline from scratch and an improved initial policy can be expected. Some pieces of literature also suggest that using both demonstrations and real interactions with the sampling ratio automatically tuned by a prioritized replay mechanism facilitates the learning process [20]. But without sufficient demonstration data, the DRL policy may also update toward the ungrounded state-action value (or state value) and, therefore, a reasonably well control behavior cannot be guaranteed.

Unlike the initialization techniques that directly decide the initial policy behavior, reward shaping is an alternative approach that allows the exploration to be biased toward the states with high potential [21]. In this way, the problem of employing the prior knowledge is transformed into defining the potential function using demonstration data. There are many ways to encode the demonstrated state-action pairs into a meaningful potential function, but only adopting the form as the difference between the new and old state-action potential, the total order over policies can remain unchanged while significantly facilitating the learning process [22]. Considering that the potential function can also be used to initialize a DRL policy apart from being integrated to form a shaping function in a normal reward shaping process, if the potential function was formulated using supervised learning or DRL pretraining approach as discussed above, similar issues would arise.

C. Contributions

The primary objective of this article is to propose a powertrain control framework based on the DRL that is able to directly train its policy behavior in the transient real-world environment without violating safety issues. There are two originally important contributions that clearly distinguish our effort from the other pieces of literature.

1) First, although the prior knowledge can be either exploited in the offline initialization phase to provide a good "cold-start" performance or merged into the existing reward function to facilitate the online exploration process, to the authors' best knowledge, there seems to be no literature that systematically make use of the prior knowledge from multiperspective views while absorbing the essence of multidomain knowledge.
2) Second, because both the initial control actions and the following "trial and error" exploration (especially during the beginning of the learning process) are expected to behave reasonably well, not only the learning process of a DRL-based powertrain controller can be accelerated but also more importantly the powertrain controller can be placed directly in a real-world environment with a high probability not to violate the given safety issues. To guarantee that the unsafe actions are not a part of the final policy, a complementary approach that uses a shield policy that overrides the learned policy with a safe backup policy as necessary to ensure safety will be combined with the proposed algorithm. This will provide an attractive optimization direction for the future DRL-based powertrain control algorithm to build on, although no such research has been found in the pieces of literature.

The rest of the article is organized as follows. In Section II, the control problem formulation and the methodologies of the initialization and reward shaping are detailed. Section III discusses a case study results of the boost control problem for a variable geometry turbocharger (VGT) equipped diesel engine. Section IV concludes the article.

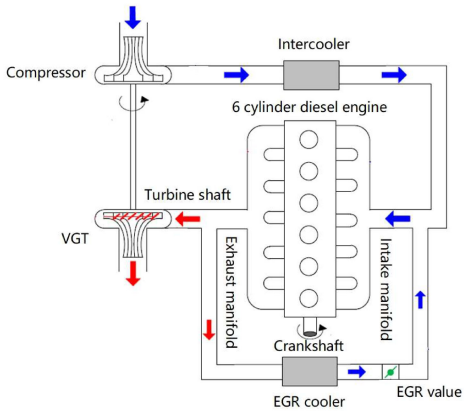

Fig. 1. VGT-equipped diesel engine with EGR system.

TABLE I
NETWORK ARCHITECTURE AND HYPERPARAMETERS SETTING

Parameters	Value
Learning rate	0.0001
Reward decay	0.9
Replay memory size	10000
Mini-batch size	128
ϵ-greedy	0.99
Number of hidden layer	3
Number of hidden layer neurons	80
Number of discrete action	17

II. METHODOLOGIES

A. Control Problem Formulation

Control problems: A learning algorithm specifically targeting the transient boost control of a VGT-equipped diesel engine in the real physical world will be proposed in this article. This control strategy will be implemented on a six-cylinder 3 L VGT-equipped engine (see Fig. 1) and aim to dynamically control the position of the VGT rack in order to track the target boost pressure. Due to the nonlinear characteristics of VGT system and the fact that exhaust gas recirculation (EGR) and VGT systems are strongly interactive, the boost control of the VGT is recognized as a major challenge for diesel engines. In this article, a control-oriented GT-Suite simulation model serves as an engine environment in order to verify the effectiveness of the proposed algorithm, but it can be applied to the real plants without modification.

DRL framework: DRL is a field of machine learning focusing on how an agent discovers which control action sequence contributes to the maximal expected cumulative return. Considering the control objective in this article and in order to measure the performance of the proposed algorithm, the cumulative return is defined as a tradeoff between the tracking accuracy and stability.

This article is based on the assumption that no high-fidelity simulation model of the engine is available but the demonstration knowledge of the control system under previous controllers and an abstract of the agent environment dynamics exist. This prior knowledge will be mined deeper so that a DRL-based VGT boost controller can be placed directly in a real-world environment for policy learning without violating safety issues. Note that this will only be done for a vanilla deep Q-network (DQN) algorithm for the demonstration but can also be extended to other DRL frameworks, such as deep deterministic policy gradient (DDPG) and soft actor critic (SAC), which will be discussed in Section III-D.

Fig. 2 shows the learning process of a DQN algorithm. For the proposed control problem, the incremental vane position of the VGT is selected as the control action (a). The four quantities, namely actual boost pressure, target boost pressure, engine speed, and current vane position, are used to form the 4-D state space (s). The reward function (r) has to be carefully designed to accurately represent the task and it is often considered as the most difficult part of an RL algorithm. Nevertheless, most of the powertrain control problems have a relatively clear idea of the reward form. For example, the control objective of this work is to track the target boost pressure under transient driving cycles by regulating the vanes in a QUICK and STABLE manner. Keeping this objective in mind, using Gaussian function as the basic function, the reward is defined as

$$r_t = e^{-\frac{[\alpha|e(t)|+\beta|I_t|]^2}{2}} \tag{1}$$

where $e(t)$ is the error between the target boost and the current boost, and I_t is the control action change rate. $e(t)$ is the term that is directly coupled to the control objective and strives to minimize the boost pressure error at every time step. I_t is designed to decrease the oscillatory behavior in the VGT control signals. The weighting α and β between the two terms depends on the requirement of the control task.

In order to break the correlations between the consecutive samples, the technique of experience replay is adopted in DQN that stores experience tuples (s, a, r, s') in the replay memory and samples uniformly at random when performing updates. Beyond that, DQN introduces a target Q-network with parameter θ^- to calculate the target. It has the same structure as the Q-network with parameter θ and the initial weights are also the same, except that the Q-network is updated every iteration, while the target Q-network is updated at regular intervals. The loss function between the target and the current Q value can be seen from 2 and it is often optimized using stochastic gradient decent. For comparison purposes, the same network architecture and hyperparameters settings, as given in Table I, are used in the following work:

$$L(\theta) = \mathbb{E}\left[\left(r + \gamma \max_{a_{i+1}} Q(s_{t+1}, a_{t+1}, \theta^-) - Q(s_t, a_t, \theta)\right)\right]. \tag{2}$$

Demonstration: The meaning of demonstration in this work is the mapping of example state to action, which derives from the system operating under an existing fine-tuned PID controller that

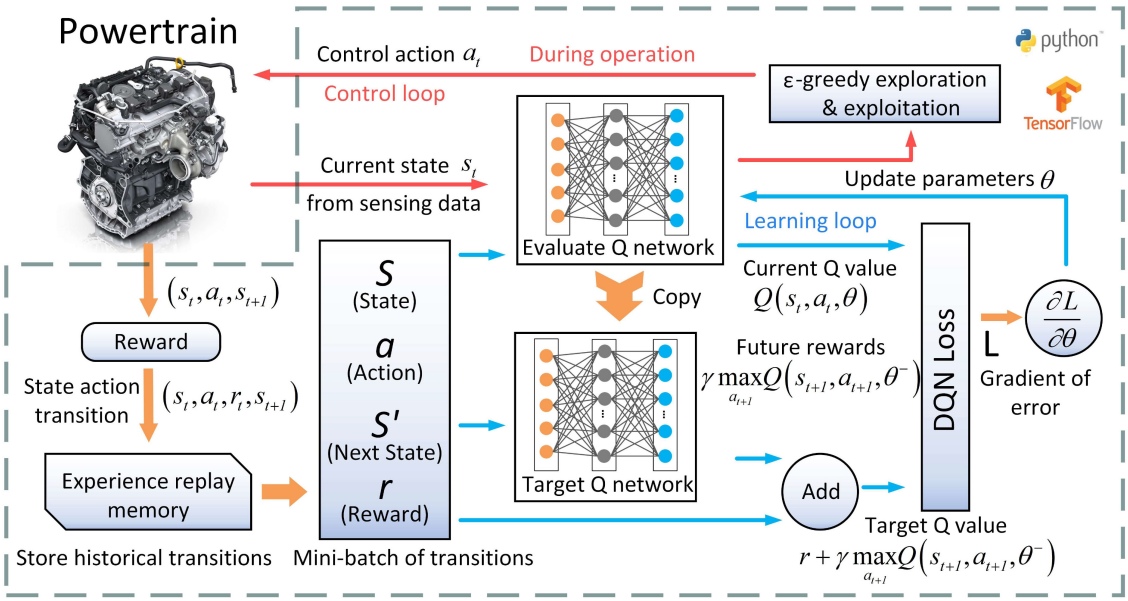

Fig. 2. Vanilla DQN implementation mechanism.

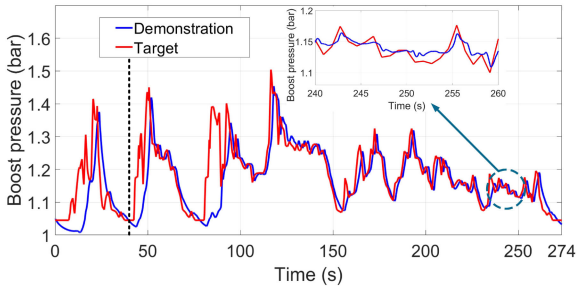

Fig. 3. Target boost pressure and the corresponding demonstration policy behavior.

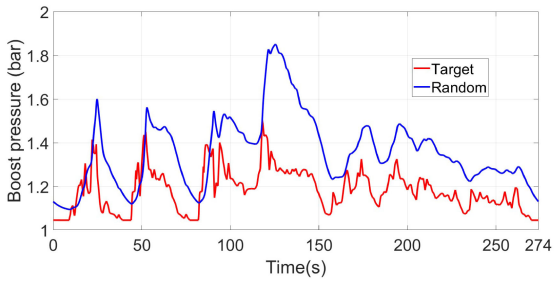

Fig. 4. Simulated boost pressure with random actions during the very first training episode.

behaves suboptimal. Fig. 3 shows the target boost pressure and the corresponding policy behavior of the demonstration under a fraction of US FTP-72 (Federal Test Procedure 72) driving cycle. Although they seem to overlap from the figure's full view, there is a gap that can be further optimized between the demonstration and the optimal from the zoom-in plot.

Exploration: In the applications of vanilla DQN, one of the random exploration techniques, namely ε-greedy procedure, has been frequently utilized to balance exploitation and exploration. In order to fully explore the state space in the simulated environment, it is a common practice to randomly select actions, i.e., ε is equal to 0 at the beginning of the training and gradually reduce randomness (while increasing the value of ε to a large number, for example 0.99) as the agent gains the experience. For industrial applications, this practice of training is strictly not allowed, as random exploration could lead the plant to some uncontrollable or even dangerous states causing irreversible losses. Taking the simulated control problems for example (see from Fig. 4), during the very first training episode when the actions were only selected randomly, there would be a large

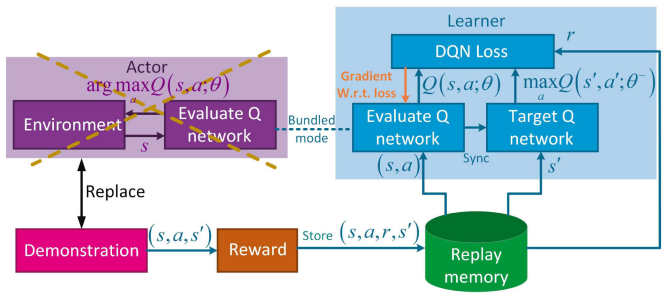

Fig. 5. DQN pretraining mechanism.

deviation between the target and the actual boost pressure. For some engine operating regions, the engine in real world could be damaged because of the very high mechanical/thermal load. Therefore, a very large value of ε is set in this article as the exploration is bias toward the mining of the prior knowledge and the purpose of small randomness is only to optimize the policy behavior steadily and compensate the system inconsistency over time (for instance, hardware aging).

B. Offline Initialization Algorithm

In this section, two different initialization techniques, namely supervised learning and DQN pretraining, are introduced in order to exploit the prior demonstration knowledge. There are many variants of supervised learning techniques and most of them work in a similar manner by the minimization of classification losses using the same network structure as the DQN policy. Following the literature [23], a score-based multiclass classification algorithm is adopted in this article [see (3)] and the neural networks are trained using a stochastic gradient descent optimization algorithm to minimize the margin classification loss:

$$J_E(Q) = \max_{a \in A} [Q(s,a) + l(a_E, a)] - Q(s, a_E) \quad (3)$$

where a_E is the action, the demonstrator selected in state s and $l(a_E, a)$ is a margin function that is 0 when $a = a_E$ and positive, otherwise. This loss forces the values of the other actions to be at least a margin lower than the value of the demonstrator's actions.

The other approach to initialize a DQN's network is via the method of pretraining that uses a similar learning process as a vanilla DQN policy. This requires the demonstration to incorporate a reward function that defines the task problems and assumes that the augmented dataset was derived from the real interactions with the environment (see Fig. 5). By doing so, the actor and the learner process of a vanilla DQN are separated and the actor process is replaced by the augmented demonstration that periodically send the quads of (s, a, r, s') to the replay memory to complete a vanilla DQN training process.

Although both the supervised learning and the DQN pretraining technique can offer a viable solution for the demonstration knowledge transfer during the initialization phase of the training, the problems inherent in the nature of them may prevent them from being widely adopted for powertrain control problems. This is mainly because the demonstration data are generally not perfect for many cases and are only covering a narrow part of the state-action space without realistic values for other unseen conditions. By coupling the theory of supervised learning and DQN pretraining [see (4)], it seems that they can back up and provide each other with ground estimates. For example, when the policy is initialized by pretraining, the network would update toward the maximum value of the next state and this can be grounded by the supervised learning simultaneously. On the contrary, the DQN update can provide the supervised learning process with the ground truth, i.e., reward function and help the Q-network satisfy the Bellman equation. This is required to improve the policy for the following fine-tuning learning process.

$$J(Q) = J_{\mathrm{DQN}}(Q) + J_E(Q). \quad (4)$$

C. Online Reward Shaping Algorithm

Apart from the practice of incorporating the prior knowledge in the initialization phase, the example demonstration can also be integrated into the standard training process by augmenting the reward expression with the potential function ϕ.

$$R_F(s, a, s') = R(s, a, s') + \phi(s, a, s'). \quad (5)$$

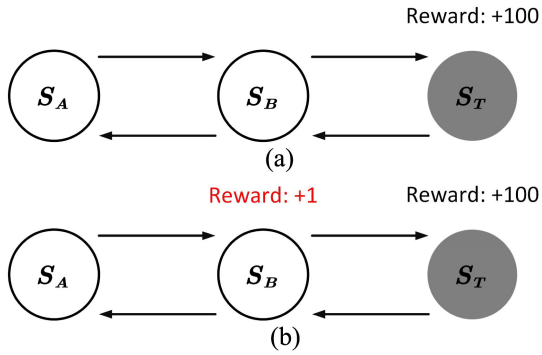

Fig. 6. Reward shaping mechanism.

The potential function is able to provide the learning controller with useful gradient information by enriching the base reward signal. It can also be formulated using the offline initialization technique described above. This is because the Q value derived from the initialization training phase is no other than the potential estimate of a state-action pair. Intuitively, this approach will bias the action exploration toward the higher augmented rewards, therefore realizing the demonstration knowledge transfer from a different angle. Nevertheless, it should be noted that modifying the reward function may change the policies that will be learned. Taking the problem in Fig. 6 for example, the agent is learning to walk from the original state S_A to the target state S_T. The reward for reaching the target state is $+$ 100 and all other nontarget rewards are 0. In the first round of learning, the agent uses random exploration strategy and the probability of being able to reach the target and get the reward is one-fourth. If the state-action space becomes larger, the probability that the agent reaches to the target for the first time will be very low. An intuitive method to solve this reward sparsity problem is to give the agent an intermediate reward, i.e., potential function in addition to the base reward function when the agent takes a step toward the goal. For example, a reward of $+1$ can be added when the agent reaching a state is closer to the target state. Although it looks effective, the optimal solution for this modified reward problem is changed to walk back and forth between S_A and S_B and get a $+1$ reward constantly, violating the original intention to only facilitate the learning process without changing the solution of the task. If we define a potential function over the state-action space and take F as the difference between the new and old state-action's potential (see the following equations), Vecerik *et al.* [20] and Brys *et al.* [21] showed that this formulation preserves the total order over policies.

$$R_F(s, a, s') = R(s, a, s') + F(s, a, s') \quad (6)$$

$$F(s, a, s', a') = \gamma \max \Phi(s', a') - \max \Phi(s, a). \quad (7)$$

Based on the discussions above, the prior knowledge can be either exploited in the offline initialization phase to provide a good "cold-start" performance or merged into the existing reward function to facilitate the online exploration process. Meanwhile, the coupling of the supervised learning and the DQN

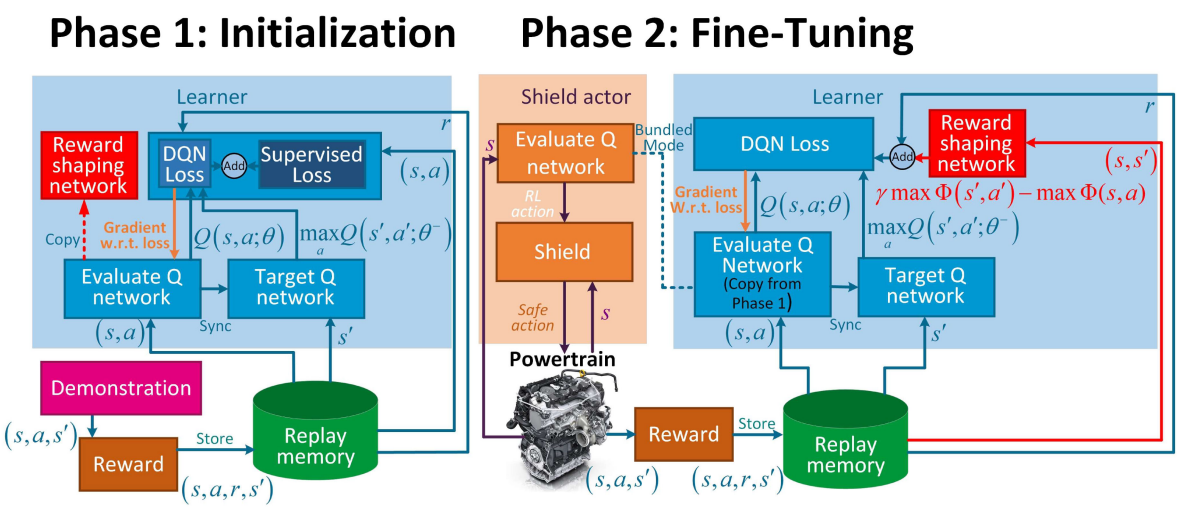

Fig. 7. Initialization and reward shaping coupling mechanism using the combination of both supervised learning and DQN pretraining.

pretraining technique can effectively utilize the prior demonstration knowledge. Thus, the next logical step is to make use of the prior knowledge from the perspective of both initialization and reward shaping using the combination of both supervised learning and standard DRL pretraining, but no such research has been found in the pieces of literature.

To guarantee that the chosen action is safe, a complementary approach that uses a shield policy will be combined with the proposed algorithm [24]. This mechanism overrides the learned policy with a safe backup policy when necessary and will allow the controller's safety guarantees and optimality to be integrated. As the shield is not the focus of this work and it is only adopted to check whether a given safety constraint is violated and guarantee the safety of the proposed algorithm just in case (which is even not activated for the proposed algorithm in Section III), a brief shield introduction is presented. In general, the shield simply constrains the set of the allowed actions in a way that ensures safety using a planning-ahead strategy. Take the control problem in this article for example, the actual boost pressure of the engine should be constrained below a prescribed level, which is determined by the mechanical and thermal loading of the engine cylinder. This safety specification can be satisfied when a planning-ahead model is given, as the system can be enforced to never cross the state's safety region by only selecting safe actions. Note that the perfect environment dynamics is not required and the coarse abstraction of the environment dynamics w.r.t the safety specification is enough to execute the shield. For example, thanks to the intuitive and mature knowledge of the reactive system between the VGT and the diesel engine, it is not difficult to decide at specific state which action is safe or not. Specifically, the shield monitors the actual boost pressure via a pressure sensor and selects safe actions accordingly, i.e., if the actual boost pressure is reaching the safety limit, the aspect ratio of the VGT is controlled smaller and vice-versa. In this article, different transient VGT behaviors at different engine speeds are considered and a conservative abstraction of the environment dynamics is built that guarantees the safety to be obeyed in real physical world. Fig. 7 shows the illustration of

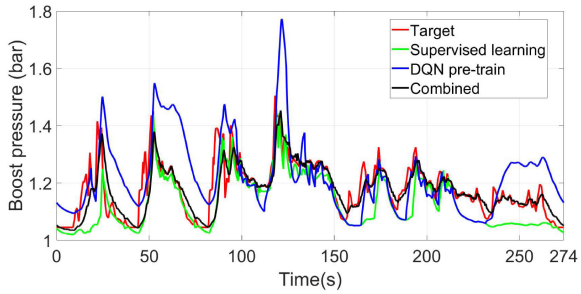

Fig. 8. Direct demonstration knowledge transfer via supervised learning, DQN pretraining, and combined.

the proposed DQN from the demonstration algorithm, together with the aforementioned shield mechanism.

III. Results and Discussions

The proposed DQN algorithm that makes use of the prior knowledge is validated in this section. In the following, first, the policy behavior directly derived from the demonstration is presented. Then, the comparisons of the following online fine-tuning learning process are made from the perspective of utilizing this demonstration via initialization and reward shaping. Finally, based on the results above, a suggestion is given on how to exploit the prior demonstration effectively and safely.

A. Offline Initialization Algorithm

In order to show the generalization capability of the knowledge transfer from the demonstration, only a small fraction of the imperfect demonstration data corresponding to the first period of boost increase and decrease is utilized (see Fig. 2 for reference). Fig. 8 shows the policy behavior directly derived from this demonstration using the technique of supervised learning, DQN pretraining, and the combined without an online fine-tuning learning process. It can be seen that the limited data, which only accounting for 14.5% of the total demonstration already, have the

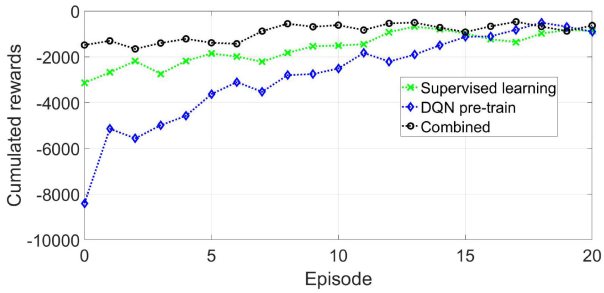

Fig. 9. Fine-tuning learning curve for supervised learning, DQN pre-training, and combined.

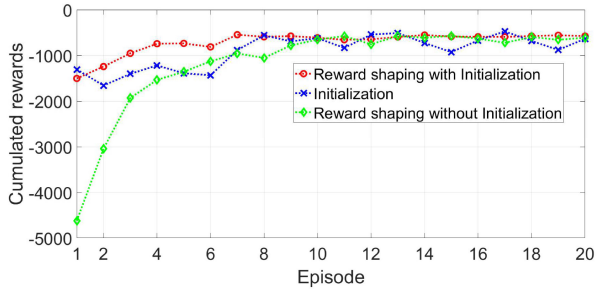

Fig. 11. Learning curve using initialization, reward shaping, and combined.

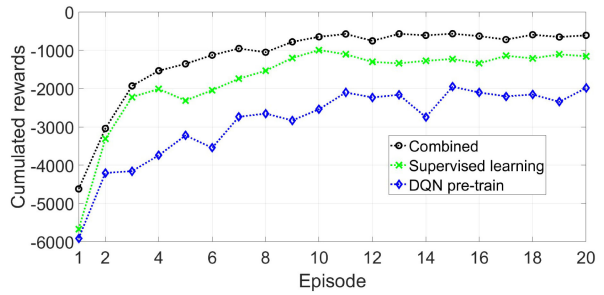

Fig. 10. Learning curve of reward shaping method using supervised learning, DQN pretraining, and combined techniques.

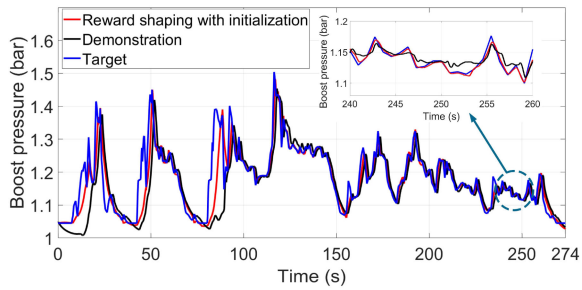

Fig. 12. Final policy behavior between the demonstration and the proposed method.

potential to generalize reasonably well and compared with the results of DQN pretraining, which has a tendency to overshoot, and that of the supervised learning, which undershoots for the most of time, the technique combining both of them performs the best.

B. Offline Initialization Algorithm Followed by Online Learning

Following the initialization phase, an online learning process is applied using a vanilla DQN exploration algorithm with constant small randomness, i.e., $\varepsilon = 0.99$. From Fig. 9, it can be seen that, although there is a big difference between the initial policy behaviors (indicated by the cumulated rewards) using the three initialization techniques, they all gradually improve to a similar level during the online learning process while interacting with the environment. Since the combined method surpasses the other two techniques during the entire online learning process, it is not difficult to draw the conclusion that the experience learned from the offline combined technique can also be beneficial for the following online learning process.

C. Online Reward Shaping Algorithm

Apart from the practice of exploiting the prior knowledge in the initialization phase of the training process, the example demonstration can also be integrated into the standard online training process by augmenting the reward expression with some potential function. In order to facilitate the comparison between the reward shaping algorithm and the aforementioned offline initialization algorithm, the potential function is also formulated using the three initialization techniques. Fig. 10 shows the learning curve of the corresponding reward shaping methods. It can be seen that they all initialize at a poor level and the reward shaping realized using the combined technique performs the best both for the initial behavior and during the following online learning process, indicating its better capability to learn in the real environment. From Figs. 9 and 10, it looks like the result of the supervised learning is better than that of the DQN pretraining. However, it is not always true for different datasets and in fact that is one of the most important reasons why the techniques of both are combined to back up each other in this article.

From the discussion above, it can be seen that the algorithm combining the techniques of both the supervised learning and DQN pretraining is able to have a better utilization of the demonstration data compared with the technique applied separately, and coupling the initialization and the reward shaping algorithms may provide good initial policy behavior and facilitate the online learning process simultaneously. This is verified in Fig. 11 in which a better "cold-start" performance and a steadier learning process have been realized using the combined technique for both the initialization and reward shaping methods using the techniques of both supervised learning and DQN pretraining.

The final policy behavior comparison between the demonstration and the proposed algorithm is made in Fig. 12. Compared with the target controller, the proposed algorithm can not only have a better final policy behavior but also feature self-adaptivity, as shown in Fig. 11, making it attractive to real plant control problems whose system consistency may not be strictly guaranteed and whose environment may change over time. Meanwhile, considering safety constraint and training efficiency, the traditional DRL algorithms often need to be trained in

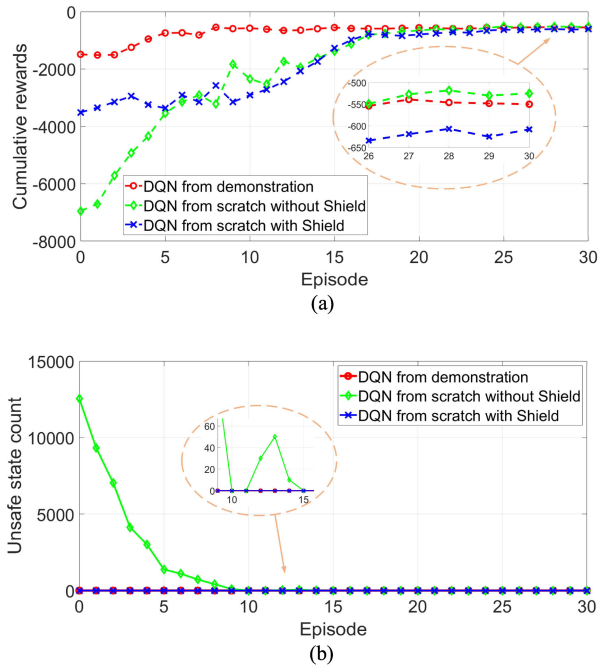

Fig. 13. (a) Learning curve (b) and unsafe state count of the DQN from scratch with and without shield and the proposed DQN from demonstration with shield.

a simulation environment and then transferred to a real physical environment. This means that the traditional DRL algorithms generally lack sufficient robustness, i.e., once a situation that does not appear in the training environment occurs in the real environment, the strategies that originally performed well in the training environment often cannot be handled correctly. The proposed modified DRL algorithm, however, has the potential to be applied directly to the real world. As there is no need to explicitly model the plant, the poor robust performance of the traditional DRL algorithms arising from discrepancies between the actual plant and its mathematical model can be greatly improved.

Finally, the proposed shield mechanism is adopted to guarantee the safety constraint to be satisfied. Fig. 13 shows the learning curve and unsafe state count of the DQN from scratch with and without shield and the proposed DQN from demonstration with shield. Compared with the vanilla DQN-based powertrain control methods requiring the control algorithm to learn from scratch, thus, cannot be placed directly in the real-world environment for training, the shielded versions of both the DQN and the proposed algorithm are able to realize safe exploration during the entire learning phase, indicating its potential to be applicable in safe critical training phase. In terms of the accumulated error, the integral absolute error of the proposed algorithm at the very first episode is improved by 74.6% compared with that of the vanilla DQN algorithm. Note that unlike the shielded DQN that guarantees the safety behavior by frequently activating the shield mechanism, the proposed algorithm by its own can constrain its action within the safety specification and the shield here is only to show whether the safety is violated or not. However, due to the fact that the shielded algorithms can only explore safe actions without full understanding of the unsafe behavior, they cannot achieve as large rewards as the vanilla DQN. This is particularly evident for the shielded DQN algorithm, while for the proposed algorithm, the last cumulated reward representing the final policy behavior is only marginally poorer than the vanilla DQN without shield. Considering the proposed algorithm improves the initial performance by 74.6% and the learning efficiency by an order of magnitude while realizing the "model-free" concept in the strict sense compared with the vanilla DQN algorithm, it is attractive for many industrial problems whose high-fidelity simulation model is unavailable or too complex to build.

D. Proposed Algorithm Under Actor-Critic (AC) DRL Framework

Although this article, taking DQN as an example, is based on the value-based DRL framework, similar techniques can also be extended to AC DRL methods, including DDPG and SAC. Specifically, in order to achieve an improved initial policy behavior, the prior knowledge can be first incorporated into the offline initialization phase using the proposed algorithm combining the techniques of both supervised learning and DRL pretraining. This is done by formulating the supervised loss in (8) and updating the actor network using the combined policy gradient with the form in (9). After that the state-action value using the initialized AC network can be considered as the potential estimate and the reward shaping algorithm that is similar with the proposed method in this work can be adopted to facilitate an effective exploration by enriching the base reward signal

$$J_E = \| a_E - \mu(s_t \mid \theta^\mu) \|^2 \tag{8}$$

$$\nabla_{\theta^\mu} J' = \nabla_{\theta^\mu} J_{\mathrm{DRL}} + \lambda \cdot \nabla_{\theta^\mu} J_E \tag{9}$$

where $\mu(s_t \mid \theta^\mu)$ and a_E represent the actor network's output with parameter θ^μ and the demonstration action, respectively. $\nabla_{\theta^\mu} J_{\mathrm{DRL}}$ is the original policy gradient. $\nabla_{\theta^\mu} J_E$ is the gradient supervised loss to the actor network weights, λ is the adjustment factor, and $\nabla_{\theta^\mu} J'$ is the combined gradient.

IV. Conclusion

In this article, a powertrain control framework based on the DQN that was able to directly train its policy behavior in transient real-world environment without violating safety issues was proposed. By incorporating the prior knowledge from previous controllers into both the offline initialization and the following online reward shaping phase while combining the techniques of both the supervised and standard DRL pretraining, taking the boost control problem for a VGT-equipped diesel engine as an example, the proposed algorithm improved the initial performance by 74.6% and the learning efficiency by an order of magnitude while meeting the safety constraint compared with its vanilla DQN-based counterpart. Considering the form of the prior knowledge in this article was easy to obtain for many industrial powertrain problems and the proposed algorithm can realize the "model-free" concept in the strict sense, it is suggested for future DRL-based powertrain control to build on.

REFERENCES

[1] R. S. Sutton and A. G. Barto, "Reinforcement learning," in *Reinforcement Learning: An Introduction*, J. Peters, Ed., 2nd ed., Cambridge, MA, USA: MIT Press, 2018, pp. 1–4.

[2] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Representation*, 2016.

[3] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[4] D. Silver *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[5] B. Hu and J. Li, "An edge computing framework for powertrain control system optimization of intelligent and connected vehicles based on curiosity-driven deep reinforcement learning," *IEEE Trans. Ind. Electron.*, to be published, doi: 10.1109/TIE.2020.3007100.

[6] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 1437–1480, 2015.

[7] O. Bastani, "Safe reinforcement learning via online shielding," 2019, *arXiv:1905.10691*.

[8] I.-M. Chen, C. Zhao, and C.-Y. Chan, "A deep reinforcement learning-based approach to intelligent powertrain control for automated vehicles," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 2620–2625.

[9] T. Liu, X. Hu, W. Hu, and Y. Zou, "A heuristic planning reinforcement learning-based energy management for power-split plug-in hybrid electric vehicles," *IEEE Trans. Ind. Inform.*, vol. 15, no. 12, pp. 6436–6445, Dec. 2019.

[10] T. Liu, Y. Zou, D. Liu, and F. Sun, "Reinforcement learning of adaptive energy management with transition probability for a hybrid electric tracked vehicle," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7837–7846, Dec. 2015.

[11] Q. Zhang, J. Lin, Q. Sha, B. He, and G. Li, "Deep interactive reinforcement learning for path following of autonomous underwater vehicle," *IEEE Access*, vol. 8, pp. 24258–24268, Jan. 2020.

[12] Q. Qi *et al.*, "Knowledge-driven service offloading decision for vehicular edge computing: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4192–4203, May 2019.

[13] H. Shi, L. Shi, M. Xu, and K.-S. Hwang, "End-to-end navigation strategy with deep reinforcement learning for mobile robots," *IEEE Trans. Ind. Inform.*, vol. 16, no. 4, pp. 2393–2402, Apr. 2020.

[14] C. Wu *et al.*, "UAV autonomous target search based on deep reinforcement learning in complex disaster scene," *IEEE Access*, vol. 7, pp. 117227–117245, Aug. 2019.

[15] H. Yang, A. Alphones, W.-D. Zhong, C. Chen, and X. Xie, "Learning-based energy-efficient resource management by heterogeneous RF/VLC for ultra-reliable low-latency industrial IoT networks," *IEEE Trans. Ind. Inform.*, vol. 16, no. 8, pp. 5565–5576, Aug. 2020.

[16] S. Zhang, Z. Sun, C. Li, D. Cabrera, J. Long, and Y. Bai, "Deep hybrid state network with feature reinforcement for intelligent fault diagnosis of delta 3-D printers," *IEEE Trans. Ind. Inform.*, vol. 16, no. 2, pp. 779–789, Feb. 2020.

[17] Z. Lőrincz, "A brief overview of imitation learning," Sep. 19, 2019. [Online]. Available: https://medium.com/@SmartLabAI/a-brief-overview-of-imitation-learning-8a8a75c44a9c

[18] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 661–668.

[19] V. de la Cruz Gabriel, Y. Du, and M. E. Taylor, "Pre-training with non-expert human demonstration for deep reinforcement learning," *Knowl. Eng. Rev.*, vol. 34, 2019, Art. no. e10.

[20] M. Vecerik *et al.*, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," 2017, *arXiv: 1707.08817*.

[21] T. Brys, A. Harutyunyan, H. B. Suay, S. Chernova, M. E. Taylor, and A. Nowe, "Reinforcement learning from demonstration through shaping," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 3352–3358.

[22] E. Wiewiora, G. Cottrell, and C. Elkan, "Principled methods for advising reinforcement learning agents," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 792–799.

[23] B. Piot *et al.*, "Boosted Bellman residual minimization handling expert demonstrations," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2014, pp. 549–564.

[24] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 2669–2678.

Bo Hu was born in Hefei, Anhui Province, China, in 1989. He received the B.S. degree from the Chongqing University of Technology (CQUT), Chongqing, China, in 2011, and the M.S. and Ph.D. degrees from the University of Bath, Bath, U.K., in 2012 and 2016, respectively, all in automotive engineering.

He is currently an Associate Professor with the Key Laboratory of Advanced Manufacturing Technology for Automobile Parts, Ministry of Education, CQUT. His current research interests include machine-learning-based control of intelligent and connected vehicles and modeling and control of advanced boosted engine systems.

Jiaxi Li was born in Xuancheng, Anhui Province, China, in 1997. He received the B.Eng. degree in automotive engineering from the Chongqing University of Technology (CQUT), Chongqing, China, in 2020.

He is currently a Research Assistant with the Key Laboratory of Advanced Manufacturing Technology for Automobile Parts, Ministry of Education, CQUT. His active research interests include machine learning, autonomous vehicle control, driver behaviors and modeling, control topics of battery, optimal control, and multiagent control.