2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Guided Online Distillation: Promoting Safe Reinforcement Learning by Offline Demonstration

Jinning Li[1] Xinyi Liu[2] Banghua Zhu[1] Jiantao Jiao[1] Masayoshi Tomizuka[1] Chen Tang[1] Wei Zhan[1]

Abstract— **Safe Reinforcement Learning (RL) aims to find a policy that achieves high rewards while satisfying cost constraints. When learning from scratch, safe RL agents tend to be overly conservative, which impedes exploration and restrains the overall performance. In many realistic tasks, e.g. autonomous driving, large-scale expert demonstration data are available. We argue that extracting expert policy from offline data to guide online exploration is a promising solution to mitigate the conserveness issue. Large-capacity models, e.g. decision transformers (DT), have been proven to be competent in offline policy learning. However, data collected in real-world scenarios rarely contain dangerous cases (e.g., collisions), which makes it prohibitive for the policies to learn safety concepts. Besides, these bulk policy networks cannot meet the computation speed requirements at inference time on real-world tasks such as autonomous driving. To this end, we propose Guided Online Distillation (GOLD), an offline-to-online safe RL framework. GOLD distills an offline DT policy into a lightweight policy network through guided online safe RL training, which outperforms both the offline DT policy and online safe RL algorithms. Experiments in both benchmark safe RL tasks and real-world driving tasks based on the Waymo Open Motion Dataset (WOMD) [1] demonstrate that GOLD can successfully distill lightweight policies and solve decision-making problems in challenging safety-critical scenarios.**

I. Introduction

Safe Reinforcement Learning (RL) aims to find a policy that not only achieves high rewards but also keeps the cost of violating constraints below a specified threshold. Traditional online safe RL algorithms [2]–[4] solve for an optimal safe policy by performing online rollouts in an environment and updating the policy accordingly. However, these algorithms always start training policies from scratch. The agent needs to learn to locate and avoid hazardous areas while it is still struggling to discover high rewards in the environment. The safety constraints discourage the agent from exploring certain hazardous areas [5], which leads to a pitfall that induces the policy to be overly conservative. The overly conservative policy often causes the agent to get stuck during its exploration, surrounded by complex hazard areas. Jammed at some states repetitively causes a skewed data distribution in the replay buffer, which deceives the policy that these states are the highest possible reward areas. It thus impedes the learning process and the overall performance.

In this situation, a near-optimal policy extracted from offline demonstrations can serve as a guide during online fine-tuning. Jump Start Reinforcement Learning (JSRL) [6], as an online fine-tuning method, has proven that training a new policy for online adaptation while using an offline extracted guide policy can be effective in regular RL settings, compared to naively initializing RL by the pre-trained policy [6]. It is natural and intuitive to propagate this meta-training scheme to the safe RL domain. The guide policy helps the agent being trained online start exploration from high-reward areas, and build new skills based on it thereafter. It is promising to save the agent from getting stuck in hazardous areas during exploration. Therefore, we propose to adapt JSRL to the safe RL setting, so that a better reward-cost trade-off can be achieved in application scenarios where the offline demonstration is available. In the safe RL literature, some prior works have investigated ways to encourage safety constraint satisfaction during training by the supervision of demonstrations [7]–[9]. In contrast, we focus on leveraging the demonstration to guide online exploration into promising areas, aiming to accelerate model-free safe RL policy training and achieve better reward-cost trade-offs.

In many real-world situations, large-scale demonstration datasets already exist [1], [10]–[14]. Prior work on imitation learning [15], [16] and offline RL [17]–[19] studies extracting policies from offline datasets. While it is seemingly promising to learn from demonstrations directly, prevalent Behavior Cloning (BC) or offline RL algorithms [20], [21] tend to fail when the demonstrations come from human experts. Decision transformer (DT) [22] has been shown as a strong method in such settings. It adopts large-scale models that are proven to have potentials [23]–[25]. Therefore, we explore the possibility of applying high-capacity decision transformers to learn from expert demonstrations in this paper. However, easily accessible datasets often lack sufficient data points in safety-critical scenarios, such as collisions in real-world traffic datasets [26], [27]. Consequently, offline datasets alone cannot provide enough information on the safety constraints, and thus offline training is not sufficient for safe RL. It strengthened the necessity of continuing to improve the decision-making policy by an online finetuning process with interactions in task environments [28], [29].

Prior work [28]–[32] typically uses the offline trained policy network architecture for online finetuning. Unfortunately, DT's transformer-based policy network, with its numerous parameters, can often fall short of meeting computation speed requirements in real-world tasks like autonomous driving. However, we do not intend to directly shrink the network size in offline training because it will sacrifice its performance to a great extent, and our experiments show that the performance and efficiency of online finetuning largely

[1] J. Li, B. Zhu, J. Jiao, M. Tomizuka, C. Tang, and W. Zhan are with the University of California, Berkeley, CA, USA. Correspondence to: Chen Tang <chen_tang@berkeley.edu>.
[2] X. Liu is with the University of Michigan, Ann Arbor, MI, USA.

rely on the quality of the offline trained guide/expert policy. Alternatively, we sought to distill a more computationally efficient policy from DT during online training. There are many prior works on network distillation [33]–[35], but their student policies are trained either by supervised learning to replicate the teacher's behavior, which isolates the student policy and the environment forbidding active explorations or by indirect ways such as reusing the critic of the teacher, which does not fully incorporate the extracted prior skills from offline demonstrations. Training a different policy with the guidance of DT instead of initializing RL with existing policy in JSRL [6] allows us to change the policy network architecture and encourage the agent to explore more promising areas, which unifies the purpose of finetuning and distillation in this paper.

In summary, we propose a training scheme, named Guided Online Distillation (GOLD), for safe RL tasks where offline expert demonstration is available. GOLD leverages an offline learned large-scale policy to guide the online learning of a computationally efficient, safe RL policy. Compared to safe RL from scratch, GOLD can improve the cumulative reward achieved by the policy while maintaining the cumulative cost below the threshold. In summary, our contributions include:

- We propose a training scheme, Guided Online Distillation (GOLD), for safety-critical scenarios where offline expert demonstration is available. It solves the problem caused by limited high-risk cases in offline datasets and conservative exploration in safe RL.
- We empirically show that adopting a DT instead of BC improves the performance of the offline extracted policy, and the large-capacity and well-performed DT guide policy is crucial for the online distilled lightweight policy to optimize its reward-cost trade-off.
- We train and evaluate the proposed algorithm on both benchmark safe reinforcement learning and real-world autonomous driving tasks extracted from the Waymo Open Motion Dataset (WOMD) [1], [10]. We show that GOLD can effectively accelerate online learning and improve policy performance.

II. Preliminaries

A. Constrained Markov Decision Process

We define a Constrained Markov Decision Process (CMDP) by a tuple $\mathcal{M} := (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, c, \gamma, \mu_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function specifying the probability $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ from state $\boldsymbol{s}_t$ to $\boldsymbol{s}_{t+1}$ when applying $\boldsymbol{a}_t$, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $c : \mathcal{S} \times \mathcal{A} \to [0, C_m]$ is the cost function for violating the constraint with C_m as the maximum cost [36], γ is the discount factor, and $\mu_0 : \mathcal{S} \to [0, 1]$ is the initial state distribution.

In safe RL, the goal is to find a policy $\pi \in \Pi$ where Π is the policy class such that it obtains a high return in reward and maintains the cost return below a threshold $\kappa \in \mathbb{R}^+$. Formally, we denote the reward value function $V_r^\pi(\mu_0) = \mathbb{E}_{\tau\sim\pi, \boldsymbol{s}_0\in\mu_0}[\sum_{t=0}^{\infty} \gamma^t r(\boldsymbol{s}_t, \boldsymbol{a}_t)]$ as the discounted cumulative reward under the policy π and the initial state distribution μ_0, where $\tau = \{\boldsymbol{s}_0, \boldsymbol{a}_0, \dots\}$ is the trajectory. The cost value function is defined similarly as $V_c^\pi(\mu_0) = \mathbb{E}_{\tau\sim\pi, \boldsymbol{s}_0\in\mu_0}[\sum_{t=0}^{\infty} \gamma^t c(\boldsymbol{s}_t, \boldsymbol{a}_t)]$. The objective is then to find an optimal policy π^* by solving the following constrained optimization problem:

$$\pi^* = \arg\max_\pi V_r^\pi(\mu_0), \text{ s.t. } V_c^\pi(\mu_0) \le \kappa. \tag{1}$$

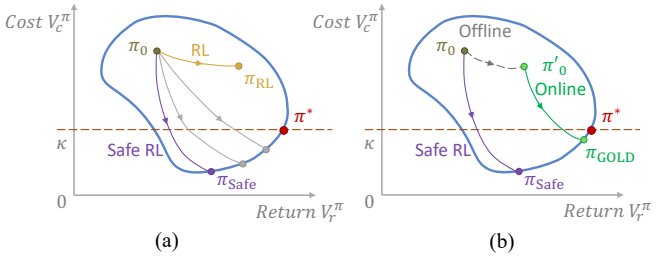

Fig. 1: An illustration of reward-cost relation for all policies in a certain environment. (a) Regular and safe RL policies converge to points far away from the optimal policy π^*. (b) Trained from offline demonstration, π'_0 is attracted toward π^*, and hence our method results in a lightweight yet more capable policy during online distillation.

B. Reward-Cost Relationship

The constraint in Eqn. 1 illustrates that a safe RL algorithm needs to control two signals simultaneously, i.e., reward and cost, compared to regular RL. A plot of cumulative cost and reward pair of policies on a $2d$-plane is informative for safe RL algorithm analysis. Given a specific environment, each policy $\pi \in \Pi$ can be mapped onto a point within the blue circle in Fig. 1, representing the reward-cost return pair (V_r^π, V_c^π) that π obtains in the environment [37]. All policies that lie below the threshold κ (the orange dash in Fig. 1) are considered feasible solutions. The optimal policy π^* obtains the highest possible reward return while maintaining the cost return below the threshold κ. When a policy is being trained in the environment, its corresponding reward-cost pair moves on this $2d$-plane. Most RL algorithms randomly initialize a policy π_0, which places it on the upper left corner in Fig. 1. If being evaluated, π_0 will obtain low reward and high cost.

Assume RL algorithms are effective in the environment. For a regular RL algorithm, e.g., PPO [38] or SAC [39], this means the reward obtained by the policy continuously increases, but there is no consideration of the cost return. This results in the yellow trajectory in Fig. 1(a). On the contrary, a safe RL algorithm [40] is dedicated to decreasing the cost and increasing reward simultaneously. Hence, its trajectory moves toward the bottom right corner in Fig. 1(a).

Once a trajectory reaches the boundary of the feasible area below κ, it stops moving because the safe RL does not allow an increase in cost or a decrease in reward. Therefore, if a safe RL algorithm penalizes too hard on cost return V_c^π, its trajectory will end up at a point on the boundary that has a low reward. The fast drop in cost during training usually leads to a convergence point π_{Safe} that is far away

from the optimal policy π^*. This aligns with our observation in experiments where safe RL agents often get stuck in a position surrounded by hazards and cannot find a way out.

In this case, offline policy extraction from expert demonstrations can provide a head start, which boosts the original initial π_0 to π_0' (closer to π^*) in Fig. 1(b). The online distillation can start from π_0' that skips exploring the environment from scratch and thus requires less effort to π_{GOLD} of higher quality than π_{Safe} even with the same online training RL backbone. Thus, we propose a new training scheme that pushes the trajectory toward the optimal policy π^* by leveraging demonstrations to extract prior skills and perform online safe RL finetuning.

III. Guided Online Distillation

In this section, we present our proposed method: Guided Online Distillation (GOLD). It consists of two stages: 1) extracting a large-scale guide policy from offline demonstration, and 2) distilling a robust but lightweight policy through online exploration with the guidance of the guide policy.

A. Extracting Expert from Offline Demonstration

Offline policy training from demonstration has been a popular research topic, and many methods have been proposed. DT [22] is an approach that lies in between BC and offline RL and proves to be competent. It adopts a similar loss function and training scheme as BC but also considers reward signals as offline RL. We therefore choose to apply DT to extract expert policies from offline demonstration and empirically show that it is superior to both BC and offline RL for safety-critical navigation and autonomous driving tasks.

The trajectory representation and model architecture follow the design in [22]. Specifically, we choose to represent the trajectory by three modalities: observation, action, and returns-to-go. Formally, the trajectory representation is $\tau = \left(\hat{R}_1, \boldsymbol{s}_1, \boldsymbol{a}_1, \ldots, \hat{R}_T, \boldsymbol{s}_T, \boldsymbol{a}_T\right)$, where $\hat{R}_t = \sum_{i=t}^{T} r_i$ is the returns-to-go, $\boldsymbol{s}_t$ and $\boldsymbol{a}_t$ are the observation and the action at time t. The model is fed with the most recent K timesteps, encompassing a total of $3K$ tokens. In the experiments in this paper, we find the default setting of $K = 20$ to be suitable for most of the tasks. A GPT [41] model processes the inputs by autoregressive modeling. Leveraging a dataset of offline trajectories, we extract minibatches with a sequence length of K from the dataset. The prediction head linked to the input token o_t is trained to predict action a_t. The loss is only evaluated on the predicted action, as no performance gain is reported by predicting observation and returns-to-go [22]. In our case, the loss is defined to be

$$L_{DT} = ||\boldsymbol{a} - \hat{\boldsymbol{a}}||^2 \tag{2}$$

where $\boldsymbol{a}$ is the ground truth action, and $\hat{\boldsymbol{a}}$ is the predicted action by the DT.

B. Online Policy Distillation

DT extracted from offline demonstration needs further distillation and finetuning in an online setting, because 1) transformers are bulk in size so not applicable to real-time systems; 2) its generalization ability is not satisfactory since the offline demonstration is not comprehensive. In GOLD, an online distillation with guided exploration by DT is proposed to train a new lightweight policy and improve its robustness on out-of-distribution hazards.

The backbone of online distillation is based on JSRL [6]. We define a guide policy as the pre-trained DT from Sec. III-A, whose parameters are frozen during online distillation. The policy network to be distilled is designed to be a lightweight Multi-Layer Perceptron (MLP). On the one hand, JSRL makes sure the reward maintains its stable improvements, instead of dropping dramatically in naive online finetuning methods. The skills of the guide DT are distilled into the lightweight policy by exposing it to a high-reward trajectory distribution. On the other hand, the environment states induced by the guide policy are also relatively safe, where the agent explores to learn fine-grained information of the costs. This makes the lightweight exploration policy not only training efficient but also robust.

The actor-critic RL training in a typical JSRL aims to approximate an optimal Q-function, which satisfies

$$Q^{\pi}(\boldsymbol{s}_t, \boldsymbol{a}_t) = r(\boldsymbol{s}_t, \boldsymbol{a}_t) + \gamma \mathbb{E}_{\boldsymbol{s}_{t+1} \sim T(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t), \boldsymbol{a}_{t+1} \sim \pi(\boldsymbol{a}_{t+1}|\boldsymbol{s}_{t+1})} \left[Q^{\pi}(\boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1}) \right].$$

In actor-critic RL, the data buffer typically only contains trajectories sampled using the single policy $\pi(\boldsymbol{a}|\boldsymbol{s})$ that is being trained concurrently. However, in the proposed online distillation stage, the trajectories of both guide and lightweight policies are saved into the data buffer. It causes a mixed and skewed data distribution, which injects too much variance into the Q-function's training process, thus hindering its prediction accuracy.

We propose to resolve the aforementioned problem by leveraging Implicit Q Learning (IQL) [17] as the training algorithm during online distillation. IQL approximates a Q-function without relying on an explicit policy by expectile regression. Specifically, it first estimates expectiles only with respect to the actions in the support of the data by first approximating a value function $V_\psi(\boldsymbol{s})$ with a loss $L_V(\psi)$,

$$L_V(\psi) = \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a}) \sim \mathcal{D}} \left[L_2^{\tau}(Q_{\hat{\theta}}(\boldsymbol{s}, \boldsymbol{a}) - V_\psi(\boldsymbol{s})) \right],$$

where $L_2^{\tau}(u) = |\tau - \mathbb{1}(u < 0)|u^2$, and τ is the expectile. It then avoids injecting stochasticity from the state distribution by averaging over the stochasticity from the dynamics transitions and fitting a Q-function Q_θ with a loss $L_Q(\theta)$,

$$L_Q(\theta) = \mathbb{E}_{(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}') \sim \mathcal{D}} \left[r(\boldsymbol{s}, \boldsymbol{a}) + \gamma V_\psi(\boldsymbol{s}') - Q_\theta(\boldsymbol{s}, \boldsymbol{a}) \right]^2.$$

The resulting Q-function is only exposed to the upper expectile of the returns in the data buffer, which makes it a better approximation of the Q-function for the optimal policy. This decoupling between the Q-function approximation and the current policy is suitable for GOLD. The learned Q-function can well predict the Q-values of the optimal policy, regardless of the mixed state distribution in the replay buffer.

Algorithm 1: Training Procedure of GOLD

Initialize: A decision transformer (DT) π_μ^g for guide policy, an lightweight policy network π_φ, a Q-network Q_θ, A target network $Q_{\bar{\theta}} = Q_\theta$, a training dataset $\mathcal{D}$, a replay buffer $\mathcal{B}$;
// Prior skills extraction from offline demonstration
for *step* n *in range(*$0,\ N$*)* **do**
  Sample a batch of b trajectory segments τ_{t-H}^t from the dataset $\mathcal{D}$;
  Update μ: $\mu_n \leftarrow \mu_{n-1} + \epsilon_\mu \nabla_\mu L_{DT}$
end
// Online distillation procedure
for *guide step* h *in* $[H_1, H_2, \ldots, H_m]$ **do**
  Assign a non-stationary policy π defined at each timestep: $\pi_{1:h} = \pi_\mu^g$, $\pi_{h+1:H} = \pi_\varphi$;
  Collect rollouts by π and append them to the replay buffer $\mathcal{B}$;
  for *train step* m *in range(*$0,\ M$*)* **do**
    Sample a batch $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$ from $\mathcal{B}$;
    Update Q_θ and π_φ by IQL;
  end
end

C. Practical Implementation

We summarize the complete proposed algorithm GOLD in Algo. 1. A DT is first trained from offline demonstration, which later serves as the guide policy during online distillation. A lightweight exploration policy network is then trained interactively in the task environment by IQL. In GOLD, the safety constraints are enforced by reward shaping, adapting IQL to safe RL settings, i.e., its actual reward is a linear combination of the original reward and cost

$$\text{reward}_{\text{new}} = \text{reward} + \lambda \cdot \text{cost},$$

since we are focused on safety-critical tasks in this paper.

IV. Experiments

A. Experiment Setting

1) Safety Gym & Bullet Safety Gym: `safety-gym` [42] and `bullet-safety-gym` [43] are open-source frameworks which are designed to train and evaluate safety performance across many tasks and environments, distinct in complexity and design. The observation of an agent is set to include the agent's own body state, the sensing information on obstacles given by pseudo laser rays, and task-specific information such as distance to goals. We pick five tasks with two different agent types. The tasks include `Circle` and `Gather`, `Goal`, `Button`, `Push`, and the agent types are `Point` and `Car`. We perform training and evaluation in different combinations of tasks and agents.

2) MetaDrive: a lightweight yet powerful driving simulator [44], which provides convenient scene composition with various road maps and traffic settings that are critical for generalizable RL. The simulation is realistic as it leverages an accurate physical engine and emulates sensory input. The driving scenes can be replayed from real-world traffic data such as WOMD [1], [10], NuScenes [11], Argoverse [12], etc. The observation consists of pseudo Lidar-like cloud points, navigation information represented by waypoints, and ego states, including steering, heading, velocity, and relative distance to boundaries. The action space contains the acceleration and steering of the ego vehicle.

3) Offline Datasets: In our problem setting, we assume access to offline datasets collected from expert demonstration. These demonstrations are near-optimal, which contain few safety-critical situations, but mostly trajectories with high reward returns. Formally, the dataset contain N expert trajectories, each of which is represented by H tuples $\{(\boldsymbol{s}_t^k, \boldsymbol{a}_t^k, \boldsymbol{s}_{t+1}^k, r_t^k)_{t=0}^T\}_{k=1}^N$, where $t \in [0, T]$ is the time step from 0 to H, and $k \in [1, N]$ is the episode number from 1 to N. In the `(bullet)-safety-gym` environments, the offline datasets are collected by expert RL policies to imitate human experts, which are trained in online settings by SAC with carefully tuned reward shaping for data collection purposes. These expert RL policies are able to reach high reward returns and satisfy cost requirements. We use Waymo Open Datasets as the offline demonstration dataset for the MetaDrive task. These datasets are recordings of real traffic scenes generated by human drivers.

B. Baselines

We compare our proposed GOLD to its own variants and state-of-the-art safe RL methods, including:

- **Safe RL Trained from Scratch**: To show the role of the guide policy in GOLD, we choose two trained-from-scratch safe RL baselines: Constrained Variational Policy Optimization (CVPO) [3] and Implicit Q-Learning (IQL) [17] equipped with reward shaping. Because of space limit, only CVPO is included as a SOTA safe RL baseline which reaches the best performance on safety gym tasks in [3]. We include IQL with reward shaping as a baseline since the nominal GOLD algorithm uses IQL during online distillation. Also, we find it works decently well even though only simple reward shaping is applied. We include it as a strong but simple baseline to encourage safe behaviors for RL policies, with the hope of inspiring the community to derive simpler but effective safe RL algorithms.
- **Variants of the proposed method**: We also demonstrate how different components in the proposed method contribute to the final performance. For the guide policy trained from offline datasets, we compare DT with BC. For the RL backbone of online distillation and training, we compare IQL with CVPO. In summary, we have four variants, namely, GOLD (BC-IQL), GOLD (DT-CVPO), and GOLD (DT-IQL).

C. Guide Policy Performance

The performance of various offline policy extraction methods in terms of both reward and cost is listed in Tab. I. The

		Offline		Online				
Task		**BC**	**DT**	**IQL**	**CVPO**	**GOLD (BC-IQL)**	**GOLD (DT-CVPO)**	**GOLD (DT-IQL)**
Car-Circle	r ↑	366.9 ± 10.4	**450.3 ± 53.8** △	630.7 ± 26.4	502.5 ± 10.8	628.4 ± 20.6	613.7 ± 26.3	**688.3 ± 4.2** □
	c ↓	41.4 ± 5.3	**40.4 ± 6.3** △	17.5 ± 2.8	7.4 ± 3.3	13.6 ± 1.4	3.9 ± 0.5	**3.2 ± 1.5** □
Car-Gather	r ↑	5.6 ± 2.31	**7.1 ± 1.52** △	10.3 ± 1.22	10.2 ± 0.87	12.8 ± 2.63	11.9 ± 0.44	**14.0 ± 1.98** □
	c ↓	0.42 ± 0.17	**0.38 ± 0.16** △	0.23 ± 0.12	0.18 ± 0.04	0.19 ± 0.27	0.15 ± 0.02	**0.14 ± 0.04** □
Point-Goal	r ↑	19.2 ± 1.4	**24.1 ± 0.5** △	31.6 ± 3.2	32.1 ± 5.3	32.1 ± 5.8	31.4 ± 3.8	**33.9 ± 6.5** □
	c ↓	16.7 ± 2.4	**15.2 ± 4.6** △	10.5 ± 3.9	8.5 ± 1.4	**8.0 ± 3.8** □	11.5 ± 2.9	8.3 ± 1.3
Point-Button	r ↑	23.7 ± 5.2	**27.8 ± 4.2** △	35.1 ± 4.7	38.2 ± 2.4	41.1 ± 3.1	39.2 ± 2.4	**44.8 ± 3.1** □
	c ↓	18.5 ± 5.9	**17.3 ± 2.5** △	8.4 ± 2.1	7.5 ± 2.8	**6.2 ± 4.2** □	6.8 ± 2.9	6.5 ± 1.1
Point-Push	r ↑	2.1 ± 3.2	**4.6 ± 2.1** △	5.3 ± 1.0	2.5 ± 0.3	6.6 ± 1.6	4.2 ± 1.1	**8.0 ± 2.3** □
	c ↓	50.2 ± 8.1	**45.4 ± 5.8** △	34.1 ± 9.1	19.3 ± 4.2	24.3 ± 3.8	**18.3 ± 4.5** □	20.4 ± 6.9
Car-Goal	r ↑	13.2 ± 1.7	**16.4 ± 3.4** △	22.8 ± 1.3	19.8 ± 1.9	27.9 ± 2.1	28.4 ± 1.2	**30.5 ± 5.4** □
	c ↓	53.4 ± 2.1	**49.9 ± 6.3** △	20.4 ± 5.2	24.3 ± 6.6	14.6 ± 2.6	12.2 ± 4.1	**10.8 ± 3.2** □
Car-Button	r ↑	19.6 ± 2.2	**26.5 ± 3.7** △	28.9 ± 10.4	27.1 ± 3.8	36.1 ± 3.0	30.5 ± 3.8	**42.0 ± 2.6** □
	c ↓	34.1 ± 10.5	**20.8 ± 9.3** △	12.5 ± 6.8	8.8 ± 1.3	9.8 ± 5.0	7.1 ± 4.3	**6.5 ± 6.1** □
Car-Push	r ↑	1.5 ± 0.1	**2.6 ± 0.2** △	3.8 ± 1.1	3.0 ± 0.4	4.5 ± 0.2	4.2 ± 0.2	**5.1 ± 0.4** □
	c ↓	65.3 ± 9.8	**58.9 ± 11.6** △	23.3 ± 2.9	19.3 ± 5.8	19.4 ± 1.9	**15.7 ± 2.3** □	17.4 ± 1.1
MetaDrive Waymo	r ↑	115.78 ± 132.89	**133.48 ± 190.50** △	26.29 ± 68.82	113.48 ± 163.84	141.93 ± 189.54	115.66 ± 163.18	**143.69 ± 175.98** □
	c ↓	**1.14 ± 1.96** △	1.25 ± 1.92	2.57 ± 2.36	1.05 ± 1.84	**1.03 ± 1.85** □	1.25 ± 2.07	1.15 ± 2.05
	sr ↑	53%	**54%** △	40%	58%	63%	62%	**73%** □

TABLE I: The performance of offline policy extraction (the best results are marked in bold and by a purple △) and online policy distillation (the best results are marked in bold and by a blue □). Metric notations are defined as, r: reward, c: cost. For the realistic driving environment based on WOMD, we also compare success rate as sr.

columns of offline methods show that DTs are superior in both reward and cost than BC in all benchmark tasks. In convention, BC or RL adopts MLP as the policy or value network. However, DTs use large models such as pre-trained GPT2 as the network backbone. This difference dramatically increases the model capacity of the policy network and thus improves the final performance by a large margin.

We observe the performance of DT is correlated with the offline dataset size. Typically, it benefits from enlarging the size of the dataset. We find it is sufficient to show the difference between DT and MLP using datasets of a scale of 100k trajectories for `(bullet-)safety-gym` tasks, and 10k trajectories from WOMD for Metadrive tasks. Due to limited hardware accelerator resources, we cannot perform more computation-intensive guide policy extraction on larger datasets. We leave it to future work as it pertains to the current trend of leveraging richer and bulkier datasets.

D. Online Distillation Evaluation

With the guide policy extracted from offline demonstrations, our proposed method finetunes and distills a lightweight yet more powerful policy network through interactions within online environments.

1) Computation Efficiency: The online training distills a much smaller policy network, i.e. a two-layer MLP with a hidden size of 256, which is standard in most RL problem settings. The number of parameters of the MLP is negligible compared to the huge transformer used by the guide policy, which usually has over 10 times more parameters (670 k in `safety-gym` tasks). The computation efficiency is thus noticeable and becomes an advantage when deploying these MLP policies compared to the huge DT, if the performance is above threshold. For the benchmark tasks in our setting, the online distilled MLP runs at 0.03 s per 100 inference runs, compared to 0.31 s per 100 inference runs of DT on a single NVIDIA GeForce RTX 2080 Ti GPU.

2) Policy Performance: The performance of all baselines and variants are listed in Tab. I under the tab "Online". Our proposed method outperforms all baselines in terms of reward and is superior in most tasks in terms of cost. We can see that algorithms with a guide in online distillation, i.e. variants of GOLD, perform better than training from scratch, i.e., IQL and CVPO. Regular safe RL algorithms tend to get stuck with local optimal solutions because they are discouraged from high-risk exploration in the environment by their cost constraints. However, with the guidance of pre-trained offline policies, GOLD avoids the exploration that leads to many failures before success and can discover highly lucrative solutions. Fig. 2 shows that the agent learns faster and better when equipped with a guide policy.

We also show the advantage of guidance during online distillation in Fig. 3. Here, the red car intends to press the yellow button on the upper right corner starting from the lower left corner, without touching the purple and blue hazardous obstacles. The red curve behind the car is its historical trajectory. The purple obstacles are moving, while the blue obstacles are staying still. In Fig. 3(a), the ego car trained with CVPO only finds a local optimal solution and chooses to avoid the bottom purple box at the beginning, which results in zigzagging trajectory and hence lower reward in the later stage of the episode. In contrast, the ego car trained with our method learns to find the most direct and safe way to the goal by the guidance of the expert policy in the early training stage and thus obtains higher reward and lower cost.

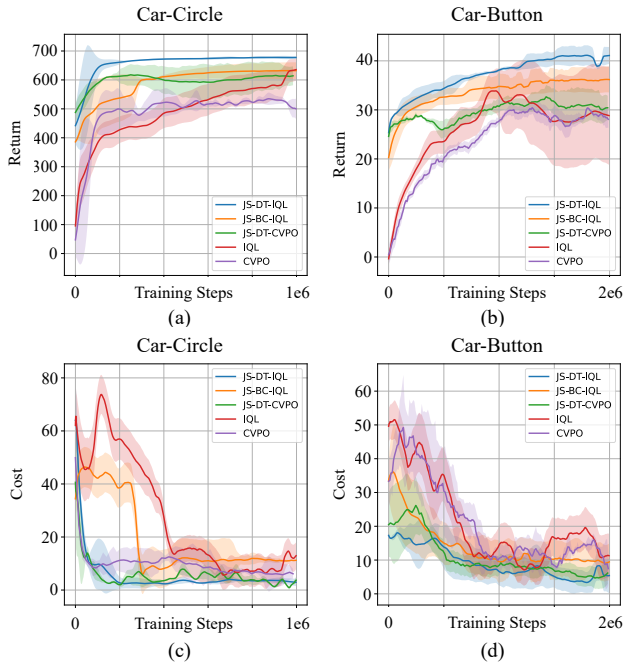

Fig. 2: The learning curves of GOLD (DT-IQL) and baselines. (a)(c): The reward and cost curves in Car-Circle. (b)(d): The reward and cost curves in Car-Button.

In Tab. I, we also show that the online distillation performance improves with a better guide policy. The algorithms GOLD (DT-*) usually obtain better rewards compared to GOLD (BC-IQL). This aligns with the intuition that a better teacher reduces the effort to learn the same level of skills.

Plus, GOLD (DT-IQL) typically performs better than GOLD (DT-CVPO), as shown in Tab. I and Fig. 2, even though they both adopt DT as the guide policy. This is because of the data distribution mismatch in the replay buffer, which is mentioned in Sec. III-B. CVPO learns Q-functions evaluating only the current explore policy being trained, which mismatches the trajectories collected by a mixture of guide and explore policies. Using IQL as the online finetuning backbone solves this problem because it learns Q-functions only by evaluating the optimal policy, which in theory, can learn from data collected by any policy.

E. Realistic Experiments in Driving Scenarios

Our method is applicable to and effective in realistic scenarios, which we demonstrate by experiments on MetaDrive. These experiments are fairly close to real-world scenes because we make MetaDrive replay vehicle trajectories from WOMD. The observations input to the ego agent, including Lidar cloud points, navigation information, and ego states, also resemble the real-world setting. The goal of the ego vehicle is to arrive at a specific target position defined in WOMD. We randomly choose 10k scenarios from WOMD for training and 1k scenes for testing.

As shown in Tab. I, our method surpasses baselines by around 15% in reward and maintains the cost below the threshold. The success rate is increased by 12% compared to the best variant, which shows GOLD is capable in realistic driving tasks. The evaluation results and analysis on previous benchmarks in Sec. IV-D are transferrable to realistic tasks, which confirms the performance of GOLD is correlated to the guide policy quality. This supports and justifies our design of offline policy extraction by DT.

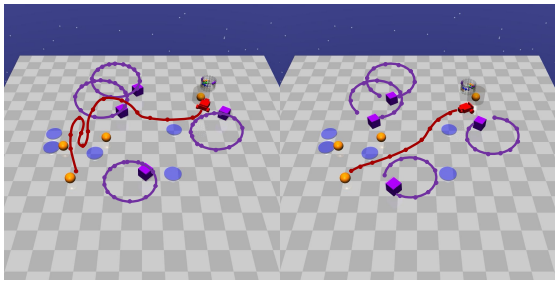

Fig. 3: Sampled trajectories in the Car-Button task. (a): GOLD (BC-IQL). The red ego car avoids the moving purple hazardous obstacle at first and struggles to find its way in the later stage. (b): GOLD (DT-IQL). The ego car can find an efficient trajectory to reach the goal, thanks to the prior knowledge inherited in the guide DT policy.

The driving experiments further validate that bringing in prior skills during online distillation is necessary for learning high-quality policy in real-world safety-critical scenarios. CVPO or IQL from scratch is too conservative to explore because it is almost impossible to discover useful skills without severe cost violations. GOLD skips the risky exploration in this safety-critical environment. With the offline trained DT as guidance, GOLD can distill and improve a lightweight policy network without struggling to jump out of the most hazardous areas. Also, CVPO outperformed IQL by a large margin when trained from scratch, but GOLD (DT-IQL) surpasses GOLD (DT-CVPO). This confirms that IQL's decoupling of Q-function and policy training works seamlessly with GOLD. More video demonstrations can be found on `https://sites.google.com/view/guided-online-distillation`.

V. Conclusion

We propose a new offline-to-online training scheme named Guided Online Distillation for safety-critical tasks. A large-scale guide policy is first extracted from offline demonstrations. It serves as a guide for online distillation, where a lightweight policy is distilled through interactions with the task environment. This lightweight network can meet computation speed requirements in realistic settings, in contrast to the bulk guide policy. The guided distillation saves the policy from being repeatedly exposed to hazards during its exploration to find useful skills, which improves its training efficiency and final performance. Experiments in both benchmarks and real-world driving experiments based on the WOMD show that the distilled policy by GOLD surpasses safe RL baselines that are trained from scratch.

REFERENCES

[1] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9710–9719, October 2021.

[2] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *International conference on machine learning*, pp. 22–31, PMLR, 2017.

[3] Z. Liu, Z. Cen, V. Isenbaev, W. Liu, S. Wu, B. Li, and D. Zhao, "Constrained variational policy optimization for safe reinforcement learning," in *International Conference on Machine Learning*, pp. 13644–13668, PMLR, 2022.

[4] J. Li, L. Sun, J. Chen, M. Tomizuka, and W. Zhan, "A safe hierarchical planning framework for complex driving scenarios based on reinforcement learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2660–2666, IEEE, 2021.

[5] S. S. Shperberg, B. Liu, and P. Stone, "Relaxed exploration constrained reinforcement learning," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pp. 2821–2823, 2023.

[6] I. Uchendu, T. Xiao, Y. Lu, B. Zhu, M. Yan, J. Simon, M. Bennice, C. Fu, C. Ma, J. Jiao, *et al.*, "Jump-start reinforcement learning," in *International Conference on Machine Learning*, pp. 34556–34583, PMLR, 2023.

[7] B. Thananjeyan, A. Balakrishna, U. Rosolia, F. Li, R. McAllister, J. E. Gonzalez, S. Levine, F. Borrelli, and K. Goldberg, "Safety augmented value estimation from demonstrations (saved): Safe deep model-based rl for sparse cost robotic tasks," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3612–3619, 2020.

[8] B. Thananjeyan, A. Balakrishna, S. Nair, M. Luo, K. Srinivasan, M. Hwang, J. E. Gonzalez, J. Ibarz, C. Finn, and K. Goldberg, "Recovery rl: Safe reinforcement learning with learned recovery zones," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4915–4922, 2021.

[9] M. Turchetta, A. Kolobov, S. Shah, A. Krause, and A. Agarwal, "Safe reinforcement learning via curriculum induction," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12151–12162, 2020.

[10] K. Chen, R. Ge, H. Qiu, R. Ai-Rfou, C. R. Qi, X. Zhou, Z. Yang, S. Ettinger, P. Sun, Z. Leng, M. Mustafa, I. Bogun, W. Wang, M. Tan, and D. Anguelov, "Womd-lidar: Raw sensor dataset benchmark for motion forecasting," *arXiv preprint arXiv:2304.03834*, April 2023.

[11] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11621–11631, 2020.

[12] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, *et al.*, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8748–8757, 2019.

[13] J. Li, H. Ma, Z. Zhang, J. Li, and M. Tomizuka, "Spatio-temporal graph dual-attention network for multi-agent prediction and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10556–10569, 2021.

[14] Z. Wu, W. Lian, C. Wang, M. Li, S. Schaal, and M. Tomizuka, "Primlafd: A framework to learn and adapt primitive-based skills from demonstrations for insertion tasks," *arXiv preprint arXiv:2212.00955*, 2022.

[15] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[16] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 4693–4700, IEEE, 2018.

[17] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," *arXiv preprint arXiv:2110.06169*, 2021.

[18] J. Li, C. Tang, M. Tomizuka, and W. Zhan, "Hierarchical planning through goal-conditioned offline reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10216–10223, 2022.

[19] J. Li, C. Tang, M. Tomizuka, and W. Zhan, "Dealing with the unknown: Pessimistic offline reinforcement learning," in *Conference on Robot Learning*, pp. 1455–1464, PMLR, 2022.

[20] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," *arXiv preprint arXiv:2108.03298*, 2021.

[21] A. Kumar, J. Hong, A. Singh, and S. Levine, "When should we prefer offline reinforcement learning over behavioral cloning?," *arXiv preprint arXiv:2204.05618*, 2022.

[22] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," *Advances in neural information processing systems*, vol. 34, pp. 15084–15097, 2021.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[24] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[25] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International Conference on Machine Learning*, pp. 8821–8831, PMLR, 2021.

[26] W. Ding, M. Xu, and D. Zhao, "Cmts: A conditional multiple trajectory synthesizer for generating safety-critical driving scenarios," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4314–4321, IEEE, 2020.

[27] J. Li, L. Sun, W. Zhan, and M. Tomizuka, "Interaction-aware behavior planning for autonomous vehicles validated with real traffic data," in *Dynamic Systems and Control Conference*, vol. 84287, p. V002T31A005, American Society of Mechanical Engineers, 2020.

[28] A. Nair, A. Gupta, M. Dalal, and S. Levine, "Awac: Accelerating online reinforcement learning with offline datasets," *arXiv preprint arXiv:2006.09359*, 2020.

[29] M. Nakamoto, Y. Zhai, A. Singh, M. S. Mark, Y. Ma, C. Finn, A. Kumar, and S. Levine, "Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning," *arXiv preprint arXiv:2303.05479*, 2023.

[30] X. Zhang, C. Wang, L. Sun, Z. Wu, X. Zhu, and M. Tomizuka, "Efficient sim-to-real transfer of contact-rich manipulation skills with online admittance residual learning," in *7th Annual Conference on Robot Learning*, 2023.

[31] C. Wang, Y. Zhang, X. Zhang, Z. Wu, X. Zhu, S. Jin, T. Tang, and M. Tomizuka, "Offline-online learning of deformation model for cable manipulation with graph neural networks," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5544–5551, 2022.

[32] Y. Sun, W. L. Ubellacker, W.-L. Ma, X. Zhang, C. Wang, N. V. Csomay-Shanklin, M. Tomizuka, K. Sreenath, and A. D. Ames, "Online learning of unknown dynamics for model-based controllers in legged locomotion," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8442–8449, 2021.

[33] A. A. Rusu, S. G. Colmenarejo, C. Gulcehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell, "Policy distillation," *arXiv preprint arXiv:1511.06295*, 2015.

[34] W. M. Czarnecki, R. Pascanu, S. Osindero, S. Jayakumar, G. Swirszcz, and M. Jaderberg, "Distilling policy distillation," in *The 22nd international conference on artificial intelligence and statistics*, pp. 1331–1340, PMLR, 2019.

[35] R. Traoré, H. Caselles-Dupré, T. Lesort, T. Sun, G. Cai, N. Díaz-Rodríguez, and D. Filliat, "Discorl: Continual reinforcement learning via policy distillation," *arXiv preprint arXiv:1907.05855*, 2019.

[36] E. Altman, "Constrained markov decision processes with total cost criteria: Lagrangian approach and dual linear program," *Mathematical methods of operations research*, vol. 48, pp. 387–417, 1998.

[37] Z. Liu, Z. Guo, Z. Cen, H. Zhang, J. Tan, B. Li, and D. Zhao, "On the robustness of safe reinforcement learning under observational perturbations," *arXiv preprint arXiv:2205.14691*, 2022.

[38] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[39] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," 2018.

[40] P. Ladosz, L. Weng, M. Kim, and H. Oh, "Exploration in deep

reinforcement learning: A survey," *Information Fusion*, vol. 85, pp. 1–22, 2022.
[41] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.
[42] A. Ray, J. Achiam, and D. Amodei, "Benchmarking Safe Exploration in Deep Reinforcement Learning," 2019.
[43] S. Gronauer, "Bullet-safety-gym: A framework for constrained reinforcement learning," 2022.
[44] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, "Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.