2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Fast Reinforcement Learning For Optimal Control of Nonlinear Systems Using Transfer Learning

Yujia Wang, Ming Xiao, and Zhe Wu

Abstract— **Traditional reinforcement learning (RL) methods for optimal control of nonlinear processes often face challenges such as substantial demands on computational resources and training time, and the difficulty of ensuring the safety of the closed-loop system during training. To overcome these challenges, this work proposes a safe transfer reinforcement learning (TRL) framework. The algorithm leverages knowledge obtained from pre-trained source tasks to expedite learning in a new yet related target task, thereby significantly reducing both learning time and computational overhead for optimizing a control policy. Additionally, the proposed TRL method collects data and optimizes the control policy within a control invariant set (CIS) to ensure the safety of the system throughout the learning process. Furthermore, we develop a theoretical analysis for the TRL algorithm that establishes an error bound between the approximate control policy and the optimal ones, accounting for the discrepancy between the target and source tasks. Finally, we validate our approach using an example of optimal control of a chemical reactor, showcasing its effectiveness in solving the optimal control problem with improved computational efficiency and safety guarantees.**

I. Introduction

Reinforcement learning (RL), a powerful paradigm for learning-based control, offers an effective solution to the challenge of optimal control in complex, nonlinear process systems. However, during the process of learning the optimal control policy, the system safety cannot be guaranteed, as RL may explore new actions or states to learn more about its environment, leading to unintended and unsafe consequences.

In addressing critical safety concerns in industrial applications, such as self-driving cars, robotics, and chemical processes, various RL strategies based on Lyapunov stability theory have been developed [1], [2]. Another branch of RL algorithms uses the Hamilton-Jacobi-Bellman (HJB) equation to derive control policies, iteratively optimizing them by updating the value function approximated with neural networks (NNs) [3], [4]. These safe RL algorithms require a considerable amount of training time to obtain an optimal control policy, as this process relies on data and experience gained through interaction with the environment. Due to these challenges, research efforts are increasingly focused on improving traditional safe RL algorithms, with transfer learning (TL) emerging as a crucial strategy in this field. Specifically, TL focuses on transferring the knowledge and experience gained from one task to another related task. The advantage of this method lies in its ability to provide a warm start to RL such that learning time is reduced and learning performance is improved.

Despite its success in supervised learning [5], TL in the realm of RL is still in its infancy. Ref. [6] highlights the benefits of TL in RL by improving agent learning processes and summarizes TL-based RL research focusing on the MDP framework in the last decade. However, a systematic analysis of the performance of TL in RL has not been established. In statistical learning theory, generalization error is widely used to assess the capabilities of various machine learning strategies and can be employed for theoretical analysis of the performance of trained models [7]. While there is research on the generalization error of neural networks in the RL domain, it focuses mainly on cases where testing and training data originate from the same distributions [4]. However, due to the discrepancy between target and source tasks, the knowledge transferred to the target process may cause performance degradation. Therefore, it is important to develop a theoretical analysis that characterizes the performance of TRL by accounting for the differences between the source and target tasks.

Motivated by the aforementioned challenges, this article develops a safe TRL framework to address the optimal control problem in nonlinear processes while ensuring system safety and improving computational efficiency. Our main contributions are as follows: 1) TL is integrated into RL to reduce the learning time for the optimal control policy; 2) the proposed TRL scheme ensures that the control policies stabilize the system with safety guarantees; and 3) a theoretical analysis for the generalization error of neural network models is developed accounting for the discrepancy distance between source and target data.

II. Preliminaries

A. Description of nonlinear systems

Consider a deterministic system described by the following continuous-time nonlinear form:

$$\dot{x} = f(x) + g(x)u \tag{1}$$

with input vector $u \in \mathbf{U} \subset \mathbf{R}^{n_u}$ and state vector $x \in \mathbf{R}^{n_x}$. The functions $f(x) : \mathbf{R}^{n_x} \to \mathbf{R}^{n_x}$ and $g(x) : \mathbf{R}^{n_x} \to \mathbf{R}^{n_x \times n_u}$ are continuously differentiable. $f(x)$ is supposed to satisfy the condition $f(0) = 0$ such that the origin $(x, u) = (0, 0)$ is an equilibrium of Eq. 1. This assumption is reasonable, as any system with non-zero equilibrium can be transformed into one with an equilibrium point at zero by using coordinate transformation. Let $\pi(x)$ represent a

This work was supported by A*STAR PIPS project (M23B3a0016) and MOE Tier 1 (22-5367-A0001).

Yujia Wang, Ming Xiao and Zhe Wu are with the Department of Chemical and Biomolecular Engineering, National University of Singapore, 117585, Singapore. (Corresponding author: Zhe Wu. Emails: wuzhe@nus.edu.sg)

control policy, signifying a mapping from state $x(t)$ to the corresponding control input $u(t)$.

Assumption 1: There exists a continuous control policy $\pi(x(t))$, such that the control input $u(t) = \pi(x(t))$ asymptotically stabilizes the system of Eq. 1 within a compact set Ω.

Definition 1: For a dynamic system, a set $\mathbf{S}$ is defined as a control invariant set (CIS) if, for any initial state within this set, there exists a control input that ensures that the state of the system remains within this set for all future times $t \geq 0$.

Considering a control policy $\pi(x)$, let its associated infinite horizon value function be defined as follows:

$$V_\pi(x(t)) = \int_t^\infty r(x(\tau), u(\tau))d\tau, \tag{2}$$

where $u(\tau) = \pi(x(\tau))$. The function $r(x(\tau), u(\tau))$ is generally designed as a quadratic, normalized cost $r(x(\tau), u(\tau)) = x^T Q x + u^T R u$, in which Q and R are positive-definite, each appropriately dimensioned to match the sizes of x and u respectively. However, depending on the control problem, $r(x(\tau), u(\tau))$ can be designed differently. For example, when considering obstacle avoidance, the cost function r might include terms that penalize proximity to obstacles. The aim of this work is to find a safe control policy $\pi(x)$ that achieves asymptotic stabilization of the system described by Eq. 1, while simultaneously minimizing the value function specified in Eq. 2.

Remark 1: Safe RL involves maximizing/minimizing the rewards of agents while adhering to safety constraints, which are defined differently from various perspectives, as discussed in Ref. [8]. In this work, safety refers to maintaining the system of Eq. 1 within a CIS for $\forall t \geq 0$, under a control policy, since instability may lead to uncontrolled events or other adverse consequences, such as reduced efficiency or increased safety hazards.

B. Background of RL-based optimal control

An RL system consists of two primary components: the environment (system) and the agent (control policy). The agent interacts with the environment through generated actions, obtaining rewards for its decisions. These rewards, in turn, guide the decisions made by the agent during subsequent action selections. The procedure is executed continuously until the optimal control policy is achieved, under the assumption of the existence of such an optimal controller. We define the optimal value function, represented by V^*, in the following manner:

$$V^*(x) = \min_{u(t)} \left\{ \int_t^\infty r(x(\tau), u(\tau))d\tau \right\} \tag{3}$$

This value function satisfies the Hamilton-Jacobi-Bellman (HJB) equation, which is expressed as:

$$\min_{u(t)} H(x, u, V^*) = \min_{u(t)} \left\{ r(x, u) + \frac{\partial V^*}{\partial x^T}(f(x) + g(x)u) \right\} = 0 \tag{4}$$

where $\frac{\partial V^*}{\partial x^T}$ denotes the partial derivation of V^* with respect to the system state x. Subsequently, by solving the following equation, the optimal control policy is derived

$$\pi^* = \arg\min_{u(t)} H(x, u, V^*) \tag{5}$$

Due to the unavailability of the optimal value function V^* , it is necessary first to construct a neural network (NN) for approximating the value function, in order to approximate the optimal control policy from the above equation. Through the policy iteration algorithm, this function and the corresponding control policy are iteratively optimized. The process continues until they approximate their optimal values within a predetermined threshold. In this context, the NN used for approximating the value function is termed the 'critic NN'. This NN is optimized by minimizing a loss function, which is formulated as a mean square error.

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left(r(x(t_i), u(t_i)) + \frac{\partial V_w}{\partial x^T}(f(x(t_i)) + g(x(t_i))u(t_i)) \right)^2 \tag{6}$$

where N represents the number of samples collected for training the critic NN. We use $V(x, t, w)$ to denote the critic NN, where w is the NN parameter vector. For simplicity, V_w is short for $V(x, t, w)$. $x(t_i)$ and $u(t_i)$ denote the state vector and input vector in the t_ith time step. The parameter vector of the critic NN is updated using the following rule.

$$w \leftarrow w - \eta \nabla_w \mathcal{L} \tag{7}$$

In this equation, η is a positive constant denoting the learning rate, and $\nabla_w \mathcal{L}$ denotes the gradient of the loss function $\mathcal{L}$, which is given by $\nabla_w \mathcal{L} = \frac{\partial \mathcal{L}}{\partial w}$. The policy $\pi(x, w)$ is iteratively updated using the trained critic NN. Due to the approximation error of the critic NN, directly updating the control policy based on V_w may not guarantee the safety of the system. Furthermore, learning the optimal control policy and value function requires significant computational resources and long training time. To address these challenges, in the next section, we will develop a novel TRL algorithm.

III. Transfer Reinforcement Learning

In this section, the focus is on developing a TRL framework that addresses the optimal control problem, taking into account both computation time and safety issues.

A. Transfer reinforcement learning (TRL)

Let $\mathcal{T}$ and $\mathcal{S}$ denote the target and source distributions, respectively. The set $S = (s_1, \ldots, s_m) = ((x_1, y_1), \ldots, (x_m, y_m))$ represents m labeled data points sampled from a specific distribution, where $x_i \in R^{d_x}$ serves as the NN input, and $y_i \in R^{d_y}$ corresponds to the labeled output. In this work, the optimal value function, denoted by V^*, is referred to as the labeling function, i.e., $y_i = V^*(x_i)$. Let $V^*_{\mathcal{S}}$ represent the optimal value function in the source task, and $V^*_{\mathcal{T}}$ denote the optimal value function in the target task, respectively. The labeled data and the labeling

function are primarily introduced for theoretical analysis, facilitating the understanding of transfer learning concepts. The proposed TRL framework does not involve collecting labeled data for training the critic NN. This is due to the unavailability of the true function V^*. Instead, the critic NN is trained by minimizing the reconstructed loss function, as shown in Eq. 6. This method relies on the fact that the optimal value function and the optimal control policy satisfy the condition $H(x, \pi^*, V^*) = 0$. Additionally, the TRL algorithm is performed using independently and identically distributed datasets, which are sampled from the source distribution $\mathcal{S}$ and the target distribution $\mathcal{T}$, respectively. Since the distributions of these samples may not perfectly match the underlying distributions from which they originate, we use $\mathcal{S}_s$ and $\mathcal{T}_s$ to represent the empirical distributions characterizing the observed data points.

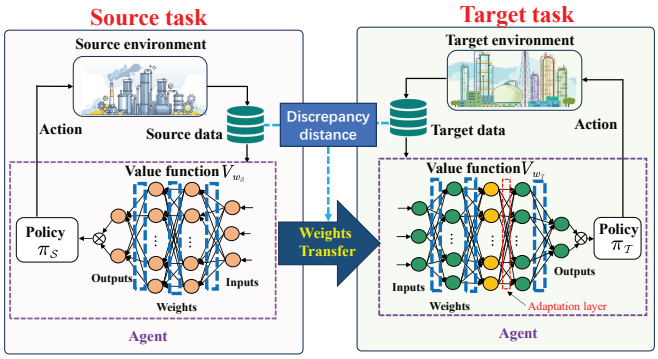

Fig. 1. Framework of the proposed TRL algorithm.

Given a source task and a target task, the TRL framework involves leveraging the knowledge learned from the source task to improve the learning efficiency of RL for solving a new but related optimal control problem in the target task. This approach is demonstrated in Fig. 1. Specifically, the optimal control policy $\pi_S(x)$ of the source task is trained using an RL algorithm with sufficient data. Generally, the control policy derived from solving the HJB Eq. 4 under stationarity conditions can be expressed as follows:

$$\pi_S(x) = -\frac{1}{2} R^{-1} g^T \frac{\partial V_S}{\partial x} \tag{8}$$

It is seen that the control policy $\pi_S(x)$ is formulated using the system dynamics $g(x)$ from Eq. 1 and the value function V. With known system dynamics, optimization of the control policy is achievable by minimizing the value function V_S. Since it is difficult to explicitly express the infinite horizon value function, a critic NN (for source task), denoted by V_{w_S}, is constructed to approximate it. However, the approximation error between V_{w_S} and V_S may jeopardize the safety of the system under the controller of Eq. 8 during the learning phase. To address this, we adopt Sontag's control law, reformulating the control policy as follows to maintain system safety within the value function V_{w_S} level set.

$$\pi_S(x) = -\Phi(x) g^T \frac{\partial V_{w_S}}{\partial x} \tag{9}$$

with

$$\Phi(x) = \frac{\frac{\partial V_{w_S}}{\partial x^T} f + \sqrt{(\frac{\partial V_{w_S}}{\partial x^T} f)^2 + \frac{\partial V_{w_S}}{\partial x^T} g R^{-1} g^T \frac{\partial V_{w_S}}{\partial x} (x^T Q x)}}{\frac{\partial V_{w_S}}{\partial x^T} g g^T \frac{\partial V_{w_S}}{\partial x}} \tag{10}$$

As demonstrated in Ref. [4], the safety of system described by Eq. 1 is guaranteed when the control policy is designed in the form of Sontag's law and the critic NN exhibits Lyapunov function properties. To ensure that the critic NN has Lyapunov function properties, the NN is constructed with its penultimate layer having the same number of neurons as there are system states. This design ensures that the output layer, calculated as the inner product of the penultimate layer's output, is non-negative for all inputs. By choosing the $tanh$ function as the activation function and setting all biases in the NN to be false, it is ensured that the origin is the unique solution of $V_{w_S}(0) = 0$, provided that the NN weights are not zero.

After developing the optimal control policy and the optimal value function for a source task, we can transfer this learned knowledge to the target task to improve learning efficiency. Since the control policy, as shown in Eq. 9, is directly related to the value function V_{w_S}, the knowledge transferable to the target task can be formulated as the information derived from the value function V_{w_S} obtained earlier. Specifically, we construct a target critic NN, denoted as V_{w_T}. It is designed with a structure identical to that of the source critic NN, V_{w_S}, but includes an additional hidden layer, i.e, 'adaptation layer', with weights initialized as an identity matrix. For the remaining layers, the weights are initialized to be the same as those of the source critic NN. This setup ensures that prior to training, the output of the target critic NN is identical to that of the source critic NN. As a result, the transfer of knowledge from the previously trained model for the source task to the target task is achieved through the transformation of NN weights. Following this transfer, we proceed to collect data from the target environment, enabling the fine-tuning of the target critic NN V_{w_T} and the optimization of the target control policy π_T.

B. Implementation of TRL

Algorithm 1 illustrates the implementation of the proposed TRL algorithm. Specifically, both a source and a target critic NN are initially initialized. It should be pointed out that, given a large number of NN weights that are randomly initialized, there is a relatively low probability that all weights will be zero. Consequently, there is a high probability that the randomly initialized critic NN can act as a Lyapunov function in a local region, as its output is positive for any non-zero input and zero for a zero input. As shown in Algorithm 1, Phase 1 involves learning the source optimal policy using the policy iteration algorithm. A dataset is collected in the CIS, and these samples are then used to train the source critic NN. The condition $H(x, \pi_S^{(k)}, V_{w_S}^{k-1}) \leq H(x, \pi_S^{(j+1)}, V_{w_S}^{j}) \leq 0$ ensures the convergence of the critic NN V_w to the optimal value function and guarantees that the time derivative of

the critic NN V_w is negative, thus ensuring the system safety under the control policy of Eq. 9 with V_w, which will be demonstrated in Theorem 1. However, training the value function V_w by minimizing Eq. 6 may not ensure the satisfaction of this condition. Therefore, we use the weights of the critic NN from the last iteration and adjust the learning rate to train the value function V_w again until this condition is met. In Phase 2, the knowledge learned from the source task is transferred to the target task by transferring the weights of the source critic NN to the target critic NN. Finally, in Phase 3, starting with the transfer learning critic NN, we learn the value function and optimize the control policy for the target task using the method illustrated in Phase 1.

Algorithm 1 TRL Algorithm

Initialize a source critic NN $V_{w_\mathcal{S}}$ with parameter vector $w_\mathcal{S}$ and a target critic NN $V_{w_\mathcal{T}}$ with parameter vector $w_\mathcal{T}$, design an appropriate learning rate η and a threshold ς.

Phase 1: *Learn the source optimal control policy.*
Set $k = 1$.
while $|V_{w_\mathcal{S}}^{(k)} - V_{w_\mathcal{S}}^{(k-1)}| \geq \varsigma$ **do**
  for $l \leftarrow 1$ **to** N_t **do**
    Randomly chosen an initial state vector x_0 in the CIS $\mathbf{S}_{w_\mathcal{S}}^{(k-1)}$
    for $t_k \leftarrow 0$ **to** t_N **do**
      Apply the control $u_\mathcal{S}(t_k) = \pi_\mathcal{S}^{(k)}(x(t_k))$ to the system of Eq. (1), and obtain the state $x(t_{k+1})$.
    end for
    Save the data (x_{t_k}, u_{t_k}), $t_k = 0, ..., t_N$ to the set $\mathbf{D}_\mathcal{S}$.
  end for
  Set $j = 1$, and $\eta_j = \eta_0$.
  while $H(x, \pi_\mathcal{S}^{(k)}, V_{w_\mathcal{S}}^{k-1}) \leq H(x, \pi_\mathcal{S}^{(j+1)}, V_{w_\mathcal{S}}^{j}) \leq 0$ is not satisfied on the grid points in the set $\mathbf{S}_{w_\mathcal{S}}^{(k)}$ **do**
    $w_{\mathcal{S}j} \leftarrow w_{\mathcal{S}k}$
    $\eta_j \leftarrow \eta_{j-1}/2$
    Train $V_\mathcal{S}^{(j)}$ by minimizing the cost function of Eq. 6
    $j = j + 1$
  end while
  let $V_{w_\mathcal{S}}^{k} = V_{w_\mathcal{S}}^{j}$, and update control policy $\pi_\mathcal{S}^{(k+1)}$ using Eq. 9.
  $k = k + 1$.
end while

Phase 2: *Knowledge transfer.*
Let $w_\mathcal{T}^{(1)} = w_\mathcal{S}^{(1)}$, and initialize $w_\mathcal{T}^{(2)}$ as an identity matrix.
for $d_\mathcal{T} \leftarrow 3$ **to** $d_\mathcal{S} + 1$ **do**
  $w_\mathcal{T}^{(d_\mathcal{T})} = w_\mathcal{S}^{(d_\mathcal{T}-1)}$.
end for

Phase 3: *Learn the optimal control policy for the target using the method illustrated in phase 1.*

Additionally, it can be found from Algorithm 1 that the initial dataset collected for training the critic NN $V_w^{(1)}$ is from the CIS $\mathbf{S}_w^{(0)}$ determined by the initial control policy with the initial critic NN $V_w^{(0)}$. Although the initial critic NN has the characteristics of Lyapunov functions, as it satisfies $V_w^{(0)}(0) = 0$, and $V_w^{(0)}(x(t)) > 0$ for any $x \in \mathbf{S}_w^{(0)}/\{0\}$, the critic NN $V_w^{(0)}$ is initialized with arbitrary weights w, the region of the CIS $\mathbf{S}_w^{(0)}$ may not be large enough such that it is difficult to choose an initial state vector to generate data for performing the policy iteration algorithm. Therefore, the challenge of ensuring a sufficiently large and satisfactory initial CIS remains unresolved and is an open question in this field. Additionally, training an initial value function and an initial control policy to their optimal values is computationally expensive and extremely time consuming. Since the proposed TRL algorithm (see Algorithm 1) is capable of transferring the knowledge learned from the source task to the target task, the aforementioned challenges are addressed, which will be demonstrated in the simulation section.

C. Convergence analysis of TRL

Following the proposed TRL algorithm in Algorithm 1, we present Theorem 1 to establish its convergence property.

Theorem 1: Given the nonlinear system of Eq. 1, when the control policy $\pi^{(k)}$ and the value function $V_w^{(k)}$ are updated and optimized by employing Algorithm 1, both the policy and value function will converge to their respective optimal values as the number of iterations k increases.

Proof: Since the control policy $\pi^{(k+1)}$ is updated using $\pi^{(k+1)} = \arg\min_{u(t)} H(x, u, V_w^k)$, the following inequality holds.

$$H(x, \pi^{(k+1)}, V_w^k) \leq H(x, \pi^{(k)}, V_w^k) \tag{11}$$

According to the definition of the value function of Eq. 2, we have

$$\begin{aligned}&\int_t^\infty r(x(\tau), \pi^{(k+1)})d\tau + \int_t^\infty \frac{\partial V_w^{(k)}}{\partial x^T}(f(x) + g(x)\pi^{(k+1)})d\tau \\ &\leq \int_t^\infty r(x(\tau), \pi^{(k)})d\tau + \int_t^\infty \frac{\partial V_w^{(k)}}{\partial x^T}(f(x) + g(x)\pi^{(k)})d\tau.\end{aligned} \tag{12}$$

It follows that

$$V_{\pi+1}(x(t)) - V_w^{(k)}(t) \leq V_\pi(x(t)) - V_w^{(k)}(t). \tag{13}$$

That is,

$$V_{\pi+1}(x(t)) \leq V_\pi(x(t)) \tag{14}$$

Meanwhile, according to the condition $H(x, \pi^{(k)}, V_w^{k-1}) \leq H(x, \pi^{(k+1)}, V_w^k) \leq 0$ in Algorithm 1, the following inequality holds.

$$\begin{aligned}&V_\pi(x(t)) + V_w^{k-1}(x(\infty)) - V_w^{k-1}(x(t)) \\ &\leq V_{\pi+1}(x(t)) + V_w^k(x(\infty)) - V_w^k(x(t))\end{aligned} \tag{15}$$

Since $\lim_{t\to\infty} V_w^{k-1}(x(t)) = \lim_{t\to\infty} \int_t^\infty r(x(\tau), \pi^{(k-1)})d\tau = 0$, and $\lim_{t\to\infty} V_w^k(x(t)) = \lim_{t\to\infty} \int_t^\infty r(x(\tau), \pi^{(k)})d\tau = 0$, Eq. 14 and Eq. 15 result in

$$0 \geq V_{\pi+1}(x(t)) - V_\pi(x(t)) \geq V_w^k(x(t)) - V_w^{k-1}(x(t)) \tag{16}$$

Therefore, we observe that $V_w^k(x(t)) \leq V_w^{k-1}(x(t))$. Given that the optimal solution to the Hamilton-Jacobi-Bellman (HJB) equation is unique, we can infer that $V_w^k(x(t))$ monotonically decreases towards the optimal value function as the number of iterations increases. Similarly, since the control policy is updated using the rule $\pi^{(k+1)} = \arg\min_{u(t)} H(x, u, V_w^k)$, it will also converge to its optimal value as the number of iterations increases. ■

D. Adaptation analysis

While there are studies on TL in RL, assessing the effectiveness of TRL schemes poses a significant challenge. To that end, our forthcoming analysis rigorously evaluates the generalizability of the TL-based policy. Our initial step involves determining the discrepancy between the target dataset and source datasets, an essential factor affecting the generalization performance of the pre-trained critic NN. Before illuminating this NN's generalization ability, we introduce two essential concepts: empirical Rademacher complexity [9], a metric for evaluating a machine learning model's complexity and generalizability on a specific training dataset, and discrepancy distance [10], a measure for the divergence between different data distributions, crucial for assessing a model's generalization over various datasets.

Definition 2: ([9]) Given a training dataset $S = \{s_1, ..., s_m\}$, and a function class $\mathcal{H}$ of real-valued functions h, the empirical Rademacher complexity of $\mathcal{H}$ with respect to S is defined as:

$$\hat{\mathfrak{R}}_S(\mathcal{H}) = \frac{1}{m}\mathbb{E}_\sigma\left[\sup_{h\in\mathcal{H}}\sum_{i=1}^{m}\sigma_i h(s_i)\right] \tag{17}$$

where $\sigma = \{\sigma_1, ..., \sigma_m\}$ denote independent random variables sampled from the Rademacher distribution, i.e., they take values in $\{-1, +1\}$, and satisfy $P(\sigma_i = -1) = P(\sigma_i = +1) = \frac{1}{2}$.

Definition 3: ([11]) Let $\mathcal{H}_k$, $k = 1, 2, ..., d_y$, denote the kth component of the function class $\mathcal{H}$, where d_y is the NN output dimension. Considering m samples that are independently and identically distributed, the Rademacher complexity of $\mathcal{H}_k$ satisfies the following condition:

$$\hat{\mathfrak{R}}_S(\mathcal{H}_k) \leq \frac{B_X (B_W)^d \sqrt{L+1+\log(d_x)}}{\sqrt{m}} \tag{18}$$

where d denotes the NN depth, d_x is the NN input dimension, B_X is the upper bound for the NN input, B_W represents the upper bound for the NN weight matrices, and L is the number of NN layers.

Definition 4: ([10]) Consider a function set $\mathcal{H}$ where each function h maps a set X to Y, and a loss function $\mathcal{L} : Y \times Y \to \mathbb{R}^+$, let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two distributions over X. The discrepancy distance between these distributions is defined as follows:

$$\begin{aligned}\text{disc}_\mathcal{L}(\mathcal{S}_1, \mathcal{S}_2) = \max_{h,h'\in\mathcal{H}} &|\mathbb{E}_{s\sim\mathcal{S}_1}[\mathcal{L}(h(s), h'(s))] \\ &- \mathbb{E}_{s\sim\mathcal{S}_2}[\mathcal{L}(h(s), h'(s))]|\end{aligned} \tag{19}$$

where h and h' represent any hypothesis functions from the set $\mathcal{H}$, and $\mathbb{E}_{s\sim\mathcal{S}}[\mathcal{L}(h(s), h'(s))]$ denotes the expected loss over the distribution $\mathcal{L}$.

After introducing the definitions of empirical Rademacher complexity and discrepancy distance, the upper bound of the discrepancy distance between the target and source distributions will be showcased.

Proposition 1: Given the source and target distributions $\mathcal{S}$ and $\mathcal{T}$, along with their corresponding empirical distributions $\mathcal{S}_s$ and $\mathcal{T}_s$, the following inequality is met with a probability of at least $1 - p$, where $p > 0$.

$$\begin{aligned}\text{disc}_\mathcal{L}(\mathcal{S}, \mathcal{T}) \leq &4\sqrt{2}L_l d_y \hat{\mathfrak{R}}_\mathcal{S}(\mathcal{H}_{km}) + 4\sqrt{2}L_l d_y \hat{\mathfrak{R}}_\mathcal{T}(\mathcal{H}_{km}) \\ &+ 3\left(\sqrt{\frac{\log(2/p)}{2m}} + \sqrt{\frac{\log(2/p)}{2n}}\right) \\ &+ \text{disc}_\mathcal{L}(\mathcal{S}_s, \mathcal{T}_s)\end{aligned} \tag{20}$$

Proof: The proof of this proposition is detailed in Propositions 1-4 in our previous work (Ref. [5]). To help readers understand better the derivation of Eq. 20, we briefly illustrate the key steps of the proof here. First, by using the triangle inequality, the distance between the distribution $\mathcal{S}$ and $\mathcal{T}$ satisfies the following inequality:

$$\begin{aligned}\text{disc}_\mathcal{L}(\mathcal{S}, \mathcal{T}) \leq &\text{disc}_\mathcal{L}(\mathcal{S}, \mathcal{S}_s) + \text{disc}_\mathcal{L}(\mathcal{S}_s, \mathcal{T}_s) \\ &+ \text{disc}_\mathcal{L}(\mathcal{T}_s, \mathcal{T})\end{aligned} \tag{21}$$

According to Ref. [9], the expected loss $\mathbb{E}_{s\sim\mathcal{S}}[\mathcal{L}(h(s), h'(s))]$ satisfies the following condition with probability at least $1 - p$ over the distribution $\mathcal{S}$.

$$\begin{aligned}\mathbb{E}_{s\sim\mathcal{S}}[\mathcal{L}(h(s), h'(s))] \leq &\frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(h(s_i), h'(s_i)) + 2\hat{\mathfrak{R}}_S(\mathcal{H}) \\ &+ 3\sqrt{\frac{\log(2/p)}{2m}}\end{aligned} \tag{22}$$

where m represents the number of the samples drawn from source distribution $\mathcal{S}$, the term $\frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(h(s_i), h'(s_i))$ is the loss computed on the sample distribution $\mathcal{S}_s$, and it can be denoted as $\mathbb{E}_{s\sim\mathcal{S}_s}[\mathcal{L}(h(s), h'(s))]$. Subsequently, according to Definition 4, and by moving the term $\frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(h(s_i), h'(s_i))$ to the left side of the above inequality, the following inequality is established.

$$\text{disc}_\mathcal{L}(\mathcal{S}, \mathcal{S}_s) \leq 2\hat{\mathfrak{R}}_S(\mathcal{H}) + 3\sqrt{\frac{\log(2/p)}{2m}} \tag{23}$$

Similarly, we have

$$\text{disc}_\mathcal{L}(\mathcal{T}, \mathcal{T}_s) \leq 2\hat{\mathfrak{R}}_\mathcal{T}(\mathcal{H}) + 3\sqrt{\frac{\log(2/p)}{2n}} \tag{24}$$

where n represents the number of samples drawn from the target distribution $\mathcal{T}$. Let $\hat{\mathfrak{R}}_\mathcal{T}(\mathcal{H}_{km})$ and $\hat{\mathfrak{R}}_\mathcal{S}(\mathcal{H}_{km})$ denote the maximum value of $\hat{\mathfrak{R}}_\mathcal{T}(\mathcal{H}_k)$ and $\hat{\mathfrak{R}}_\mathcal{S}(\mathcal{H}_k)$, respectively. By using contraction inequality [12], the upper bound of $\hat{\mathfrak{R}}_\mathcal{S}(\mathcal{H})$ and $\hat{\mathfrak{R}}_\mathcal{T}(\mathcal{H})$ are derived.

$$\hat{\mathfrak{R}}_\mathcal{T}(\mathcal{H}) \leq 2\sqrt{2}L_l d_y \hat{\mathfrak{R}}_\mathcal{T}(\mathcal{H}_{km}) \tag{25}$$

$$\hat{\mathfrak{R}}_\mathcal{S}(\mathcal{H}) \leq 2\sqrt{2}L_l d_y \hat{\mathfrak{R}}_\mathcal{S}(\mathcal{H}_{km}) \tag{26}$$

where L_l denotes the Lipschitz constant of the loss function $\mathcal{L}$. By using Eqs. 22-26, Eq. 21 can be rewritten as Eq. 20. ■

Remark 2: In practice, there could be multiple source tasks that have similar configurations as the target task and can be used to generate a large training dataset for pre-trained models. A pre-trained model trained with a multi-source dataset may learn patterns of multiple source processes, and thus, provide a robust starting point for learning the new task to enhance its exploitation ability. In this case, multi-source transfer learning techniques can be integrated into the proposed transfer RL framework to guide the selection of source tasks and the construction of pre-trained models to achieve the best performance of adaptation [13].

Remark 3: Eq. 20 can be used to evaluate the distance between target distribution $\mathcal{T}$ and the source distribution $\mathcal{S}$. This distance can be used to measure the similarity between the source environment and the target environment. When there is a high similarity, the critic NN and the resulting control policy obtained for the source environment can significantly improve the performance of the critic NN and control policy for the target environment. Specifically, they can provide a larger CIS for the target task, and significantly reduce the training time. Additionally, this distance can help choose a source with a small distance from the target source to guarantee satisfactory transferability of the TRL algorithm.

We will next develop a theorem to demonstrate the generalizability of the critic NN in the target task after assessing the distance between the target and source distributions. This will include an analysis of the discrepancy distance and other factors that influence the generalizability of the NN of the critic.

Theorem 2: Consider a function set $\mathcal{H}$ where each hypothesis h maps the critic NN inputs to its outputs, and design a loss function $\mathcal{L}$ of Eq. 6. When m samples are collected from the source distribution $\mathcal{S}$ and n samples from the target distribution $\mathcal{T}$, the equation defining the generalization error bound of the critic neural network holds true with a probability of at least $1-p$.

$$\begin{aligned}
&\mathbb{E}_{s\sim\mathcal{T}}[\mathcal{L}(h,V_{\mathcal{T}}^*)]\\
\leq&\mathcal{L}_{\mathcal{T}_s}(V_{w_{\mathcal{T}_s}}^*,V_{w_{\mathcal{S}_s}}^*)+\mathcal{L}_{\mathcal{S}_s}(V_{w_{\mathcal{S}_s}}^*,h)+\text{disc}_{\mathcal{L}}(\mathcal{T}_s,\mathcal{S}_s)\\
&+8\sqrt{2}L_l d_y\left(\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{H}_{km})+\hat{\mathfrak{R}}_{\mathcal{T}}(\mathcal{H}_{km})\right)+\mathcal{L}_{\mathcal{T}}(V_{w_{\mathcal{T}_s}}^*,V_{\mathcal{T}}^*)\\
&+6\left(\sqrt{\frac{\log(2/p)}{2m}}+\sqrt{\frac{\log(2/p)}{2n}}\right) \qquad (27)
\end{aligned}$$

Proof: Let $V_{w_{\mathcal{S}}}^*$ and $V_{w_{\mathcal{T}}}^*$ denote the optimal hypothesis on the distributions $\mathcal{S}$ and $\mathcal{T}$. By utilizing the triangle inequality property, for any hypothesis $h\in\mathcal{H}$, we have

$$\begin{aligned}
\mathcal{L}_{\mathcal{T}}(h,V_{\mathcal{T}}^*)\leq&\mathcal{L}_{\mathcal{T}}(h,V_{w_{\mathcal{S}_s}}^*)+\mathcal{L}_{\mathcal{T}}(V_{w_{\mathcal{S}_s}}^*,V_{w_{\mathcal{T}_s}}^*)\\
&+\mathcal{L}_{\mathcal{T}}(V_{w_{\mathcal{T}_s}}^*,V_{\mathcal{T}}^*) \qquad (28)
\end{aligned}$$

with

$$\begin{aligned}
\mathcal{L}_{\mathcal{T}}(V_{w_{\mathcal{T}_s}}^*,V_{\mathcal{T}}^*)\leq&\mathcal{L}_{\mathcal{T}}(V_{w_{\mathcal{T}_s}}^*,V_{w_{\mathcal{S}_s}}^*)+\mathcal{L}_{\mathcal{T}}(V_{w_{\mathcal{S}_s}}^*,h)\\
&+\mathcal{L}_{\mathcal{T}}(h,V_{\mathcal{T}}^*) \qquad (29)
\end{aligned}$$

where $V_{w_{\mathcal{S}_s}}^*$ and $V_{w_{\mathcal{T}_s}}^*$ represent the optimal hypothesis on the empirical distributions $\mathcal{S}_s$ and $\mathcal{T}_s$. Meanwhile, according to Definition 4, we have

$$\begin{aligned}
&|\mathcal{L}_{\mathcal{T}}(V_{w_{\mathcal{T}_s}}^*,V_{\mathcal{T}}^*)-\mathcal{L}_{\mathcal{T}}(h,V_{\mathcal{T}}^*|\\
&\leq\mathcal{L}_{\mathcal{T}}(V_{w_{\mathcal{T}_s}}^*,V_{w_{\mathcal{S}_s}}^*)+\mathcal{L}_{\mathcal{T}}(V_{w_{\mathcal{S}_s}}^*,h). \qquad (30)
\end{aligned}$$

Since the distributions $\mathcal{S}$ and $\mathcal{T}$ are unknown, we use the empirical distributions $\mathcal{S}_s$ and $\mathcal{T}_s$, and Eq. 30 can be further expressed as follows by using the Definition 4

$$\begin{aligned}
&|\mathcal{L}_{\mathcal{T}}(V_{w_{\mathcal{T}_s}}^*,V_{\mathcal{T}}^*)-\mathcal{L}_{\mathcal{T}}(h,V_{\mathcal{T}}^*|\\
&\leq\mathcal{L}_{\mathcal{T}}(V_{w_{\mathcal{T}_s}}^*,V_{w_{\mathcal{S}_s}}^*)+\mathcal{L}_{\mathcal{S}}(V_{w_{\mathcal{S}_s}}^*,h)+\text{disc}_{\mathcal{L}}(\mathcal{S},\mathcal{T})\\
&\leq\mathcal{L}_{\mathcal{T}_s}(V_{w_{\mathcal{T}_s}}^*,V_{w_{\mathcal{S}_s}}^*)+\text{disc}_{\mathcal{L}}(\mathcal{T},\mathcal{T}_s)+\mathcal{L}_{\mathcal{S}_s}(V_{w_{\mathcal{S}_s}}^*,h)\\
&\quad+\text{disc}_{\mathcal{L}}(\mathcal{S},\mathcal{S}_s)+\text{disc}_{\mathcal{L}}(\mathcal{S},\mathcal{T}). \qquad (31)
\end{aligned}$$

Substituting Eq. 21 into the above inequality results in:

$$\begin{aligned}
&|\mathcal{L}_{\mathcal{T}}(V_{w_{\mathcal{T}_s}}^*,V_{\mathcal{T}}^*)-\mathcal{L}_{\mathcal{T}}(h,V_{\mathcal{T}}^*|\\
&\leq\mathcal{L}_{\mathcal{T}_s}(V_{w_{\mathcal{T}_s}}^*,V_{w_{\mathcal{S}_s}}^*)+\mathcal{L}_{\mathcal{S}_s}(V_{w_{\mathcal{S}_s}}^*,h)\\
&\quad+2\text{disc}_{\mathcal{L}}(\mathcal{S},\mathcal{S}_s)+2\text{disc}_{\mathcal{L}}(\mathcal{T},\mathcal{T}_s)+\text{disc}_{\mathcal{L}}(\mathcal{T}_s,\mathcal{S}_s). \qquad (32)
\end{aligned}$$

According to Eqs. 23-26, the following inequality holds with a probability of at least $1-p$.

$$\begin{aligned}
&|\mathcal{L}_{\mathcal{T}}(V_{w_{\mathcal{T}_s}}^*,V_{\mathcal{T}}^*)-\mathcal{L}_{\mathcal{T}}(h,V_{\mathcal{T}}^*)|\\
&\leq\mathcal{L}_{\mathcal{T}_s}(V_{w_{\mathcal{T}_s}}^*,V_{w_{\mathcal{S}_s}}^*)+\mathcal{L}_{\mathcal{S}_s}(V_{w_{\mathcal{S}_s}}^*,h)+\text{disc}_{\mathcal{L}}(\mathcal{T}_s,\mathcal{S}_s)\\
&\quad+8\sqrt{2}L_l d_y\left(\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{H}_{km})+\hat{\mathfrak{R}}_{\mathcal{T}}(\mathcal{H}_{km})\right)\\
&\quad+6\left(\sqrt{\frac{\log(2/p)}{2m}}+\sqrt{\frac{\log(2/p)}{2n}}\right) \qquad (33)
\end{aligned}$$

Since $\mathbb{E}_{s\sim\mathcal{T}}[\mathcal{L}(h,V_{\mathcal{T}}^*)]=\mathcal{L}_{\mathcal{T}}(h,V_{\mathcal{T}}^*)$, we derive Eq. 27 by moving $\mathcal{L}_{\mathcal{T}}(V_{w_{\mathcal{T}_s}}^*,V_{\mathcal{T}}^*)$ to the right-hand side of Eq. 33. ■

In Eq. 27, the term $\mathcal{L}_{\mathcal{S}_s}(V_{w_{\mathcal{S}_s}}^*,h)$ denotes the hypothesis h and the expected loss between the optimal hypothesis $V_{w_{\mathcal{S}_s}}^*$ within the source empirical distribution $\mathcal{S}_s$. The term $\mathcal{L}_{\mathcal{T}_s}(V_{w_{\mathcal{T}_s}}^*,V_{w_{\mathcal{S}_s}}^*)$ denotes the expected loss between the optimal hypotheses $V_{w_{\mathcal{T}_s}}^*$ and $V_{w_{\mathcal{S}_s}}^*$, corresponding to the empirical distributions $\mathcal{T}_s$ and $\mathcal{S}_s$, respectively. The discrepancy distance between the empirical distributions $\mathcal{T}_s$ and $\mathcal{S}_s$ is denoted by the term $\text{disc}_{\mathcal{L}}(\mathcal{T}_s,\mathcal{S}_s)$. The values of these three terms can be determined after obtaining the optimal hypotheses $V_{w_{\mathcal{T}_s}}^*$ and $V_{w_{\mathcal{S}_s}}^*$. The term $\mathcal{L}_{\mathcal{T}}(V_{w_{\mathcal{T}_s}}^*,V_{\mathcal{T}}^*)$ represents the expected loss between the optimal hypotheses $V_{w_{\mathcal{T}_s}}^*$ and the optimal value function. The term $8\sqrt{2}L_l d_y\left(\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{H}_{km})+\hat{\mathfrak{R}}_{\mathcal{T}}(\mathcal{H}_{km})\right)$ is related to the output dimension d_y, the Lipschitz constant L_l of the loss function, and the Rademacher complexity, where $\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{H}_{km})$ and $\hat{\mathfrak{R}}_{\mathcal{T}}(\mathcal{H}_{km})$ have upper bounds that are related to the input and output NN dimensions of the NN, the number of NN layers, and the upper bounds for the NN input and weight matrices, respectively (as shown in Definition 3). The last term is related to the sample numbers m and n, and the confidence level $p>0$. By selecting a sufficiently large number of training samples, constructing an NN with an appropriate structure, and tuning the NN parameters, a sufficiently small generalization error can be ensured.

IV. Simulation Study

This section uses a chemical process as a case study to illustrate the efficacy of the proposed TRL Algorithm 1, with the implementation procedure detailed in Section III.

A. Process description

The chemical process under consideration is a continuous stirred tank reactor (CSTR). Specifically, reactant A is transformed into product B in the reaction. This reactor operates with a liquid volume V, receiving reactant A at a volumetric flow rate F and an inlet temperature T_0. Our goal is to regulate the concentration C_A of reactant A in the reactor and the reactor's temperature T by manipulating the feed concentration C_{A0} of A and heat input rate Q. These controls are essential for ensuring the production of the product B. The CSTR, as described in Ref. [14], is chosen as the source CSTR. The target CSTR's parameters are set at 90% of the source process's values, except for the ideal gas constant R. Below is the dynamic model for both.

$$\begin{aligned}\frac{dC_A}{dt} &= \frac{FC_{A0}}{V_L} - k_0 C_A e^{\frac{-E}{RT}} - \frac{FC_A}{V_L} \\ \frac{dT}{dt} &= \frac{Q}{\rho_L C_p V_L} + \frac{F(T_0 - T)}{V_L} - \frac{\Delta H k_0 C_A}{\rho_L C_p} e^{\frac{-E}{RT}}.\end{aligned} \tag{34}$$

Let the steady-state of the CSTR be denoted by (C_{As}, T_s), and (C_{A0_s}, Q_s). The manipulated inputs, namely the inlet concentration of A, $\Delta C_{A0} = C_{A0} - C_{A0_s}$, and the heat input rate $\Delta Q = Q - Q_s$, are subject to physical constraints. The manipulated inputs $|\Delta Q|$ and $|\Delta C_{A0}|$ are saturated by $0.167\ kJ/min$ and $5\ kmol/m^3$, respectively. We utilize deviation variables to rewrite the input and state vectors as $u = [\Delta C_{A0}, \Delta Q]^T$ and $x = [C_A – C_{As}, T – T_s]^T$, respectively. The performance value function is designed as Eq. 2 with the parameters $R = [1, 0; 0, 1]$ and $Q = [9.35, 0.41; 0.41, 0.02]$.

B. Development of TRL and simulation results

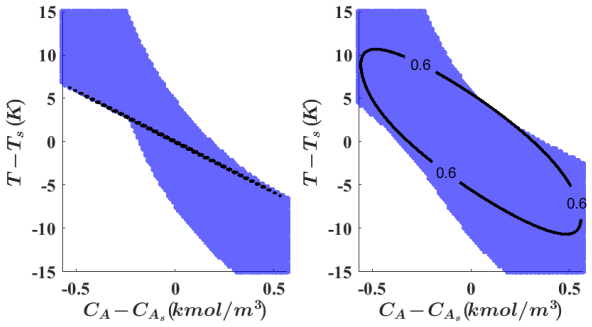

Fig. 2. CSI derived from the control policy in the source task, utilizing the initial value function and the approximate optimal value function, respectively.

Based on the description in Section III, we construct critic NNs for both source and target tasks and initialize the source critic NN with arbitrary weights. Given that the source critic NN is constructed with the properties of a Lyapunov function, it yields an initial control policy capable of stabilizing the closed-loop system. Subsequently, we compute the maximum CIS within the stability region, as shown in the left subfigure of Fig. 2. Specifically, the black ellipse represents the maximum CIS, and the blue region represents the region where the derivative of the Lyapunov function under the initial control policy is less than zero. Due to the arbitrary initialization of weights, the region of the maximum CIS is relatively small, bringing challenges in selecting initial system states from which to gather a sufficient dataset for training. We carefully select the initial points within this CIS as the initial states of the system. Applying the initial control policy to the source CSTR, we collect data from these trajectories to train the source critic NN and the control policy. By implementing Algorithm 1, the source value function and its corresponding control policy are optimized. However, as training progresses, the critic NN progressively approximates the optimal value function, consequently expanding the CIS. After 12 iterations, the approximate optimal value function and the approximate optimal control policy are derived. The largest CIS under the approximate optimal control policy is illustrated in the right subfigure of Fig. 2. It can be seen that the region of the maximum CIS within the stability region is considerably larger than that in the left subfigure.

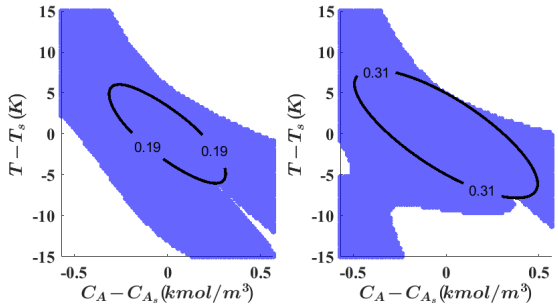

Fig. 3. CSI derived from the control policy in the target task, utilizing the initial target value function and the approximate target optimal value function, respectively.

Subsequently, we transferred the weights of the source-trained critic NN to the target critic NN. This process yields the initial target value function and its corresponding control policy. The maximum CIS under the initial target control policy within the stability region is depicted in the left subfigure of Fig. 3. It is observed that the region of the maximum CIS under the initial target control policy is noticeably larger than that under the initial source control policy. The reason for this is that the initial target control policy incorporates pretrained information based on the source CSTR. This provides a better initial control policy for the target CSTR, which also facilitates data collection for training due to the larger region of the maximum CIS. Using the Algorithm 1, after 3 iterations, the approximated optimal value function and the optimal control policy are obtained in the target task. The maximum CIS under the target approximate optimal control policy within the stability region is depicted in the right subfigure of Fig. 3. It illustrates

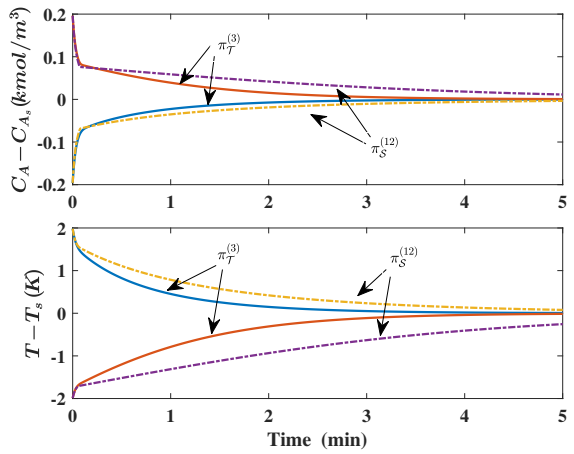

Fig. 4. State profiles ($x_1 = C_A - C_{As}$ and $x_2 = T - T_s$) for the initial conditions $(-0.2\ kmol/m^3, 2\ K)$ and $(0.2\ kmol/m^3, -2\ K)$ are presented under the initial target control policy and the approximate target optimal control policy, respectively. Specifically, the solid lines represent the trajectories under the approximate optimal control policy, while the dashed lines denote the trajectories under the initial control policy.

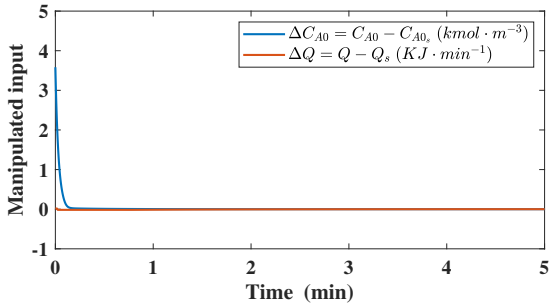

Fig. 5. Manipulated input profiles ($u_1 = \Delta C_{A0}$ and $u_2 = \Delta Q$) for the initial condition $(-0.2\ kmol/m^3, 2\ K)$ under the approximate target optimal control policy.

a significant reduction in computational load achieved by the developed TRL algorithm, as it requires only 3 iterations to derive an approximate optimal control policy in the target task, in contrast to the 12 iterations needed by the standard RL algorithm in the source task, where the control policy is optimized using arbitrarily initialized critic NN.

Fig. 4 compares the closed-loop trajectories obtained from the initial target control policy and those derived from the approximate optimal control policy developed using the TRL algorithm. The solid lines in the figure represent closed-loop trajectories using the initial target control policy prior to training in the target CSTR. In contrast, the dashed lines indicate the trajectories achieved with the approximate optimal control policy. It can be found that the trajectories are optimized by the TRL algorithm, as they converge to the origin more rapidly compared to those using the pre-training control policy. This demonstrates the effectiveness of the target approximate optimal control policy generated by the TRL algorithm. Finally, Fig. 5 shows the manipulated input profiles, which are constrained by $(5\ kmol/m^3, 0.167\ K)$, for the initial condition $(-0.2\ kmol/m^3, 2\ K)$. It is illustrated that the inputs remain within the constraints at all times under the approximate optimal control policy of the target CSTR.

V. Conclusion

This work developed a TRL scheme to solve the optimal control problem in nonlinear systems. The proposed learning method collected data within the CIS and used a control policy based on a critic NN that possessed Lyapunov function characteristics, thereby maintaining the safety of the closed-loop system during the learning phase. Furthermore, we utilized a transferred critic NN, which possesses the knowledge learned from a source process to assist in training the optimal value function and control policy for the target process, which significantly reduces computational burden and training time. Simulations of a CSTR example showed that the proposed TRL scheme maintained the safety and stability of the closed-loop system, and optimized the closed-loop control performance.

References

[1] M. Han, L. Zhang, J. Wang, and W. Pan, "Actor-critic reinforcement learning for control with stability guarantee," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6217–6224, 2020.

[2] A. S. Kumar, L. Zhao, and X. Fernando, "Task off-loading and resource allocation in vehicular networks: A lyapunov-based deep reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 10, pp. 13360–13373, 2023.

[3] D. Wang, N. Gao, M. Ha, M. Zhao, J. Wu, and J. Qiao, "Intelligent-critic-based tracking control of discrete-time input-affine systems and approximation error analysis with application verification," *IEEE Transactions on Cybernetics*, pp. 1–12, 2023.

[4] Y. Wang and Z. Wu, "Control lyapunov-barrier function-based safe reinforcement learning for nonlinear optimal control," *AIChE Journal*, p. e18306, 2023.

[5] M. Xiao, C. Hu, and Z. Wu, "Modeling and predictive control of nonlinear processes using transfer learning method," *AIChE Journal*, p. e18076, 2023.

[6] Z. Zhu, K. Lin, A. K. Jain, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 11, pp. 13344–13362, 2023.

[7] M. Xiao and Z. Wu, "Modeling and control of a chemical process network using physics-informed transfer learning," *Industrial & Engineering Chemistry Research*, vol. 62, no. 42, pp. 17216–17227, 2023.

[8] S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, Y. Yang, and A. Knoll, "A review of safe reinforcement learning: Methods, theory and applications," *arXiv preprint arXiv:2205.10330*, 2022.

[9] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[10] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," *arXiv preprint arXiv:0902.3430*, 2009.

[11] S. Chen, Z. Wu, and P. D. Christofides, "Statistical machine-learning-based predictive control using barrier functions for process operational safety," *Computers and Chemical Engineering*, vol. 163, p. 107860, 2022.

[12] A. Maurer, "A vector-contraction inequality for rademacher complexities," in *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy*, pp. 3–17, Springer, 2016.

[13] J. Hoffman, M. Mohri, and N. Zhang, "Algorithms and theory for multiple-source adaptation," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[14] Z. Wu, F. Albalawi, Z. Zhang, J. Zhang, H. Durand, and P. D. Christofides, "Control lyapunov-barrier function-based model predictive control of nonlinear systems," *Automatica*, vol. 109, p. 108508, 2019.