Cocktail: Learn a Better Neural Network Controller from Multiple Experts via Adaptive Mixing and Robust Distillation

Yixuan Wang, Chao Huang, Zhilu Wang, Shichao Xu, Zhaoran Wang, Qi Zhu
Northwestern University, Evanston, IL
{yixuanwang2024@u., chao.huang@, zhilu.wang@u., shichaoxu2023@u., qzhu@}northwestern.edu, zhaoranwang@gmail.com

Abstract—**Neural networks are being increasingly applied to control and decision making for learning-enabled cyber-physical systems (LE-CPSs). They have shown promising performance without requiring the development of complex physical models; however, their adoption is significantly hindered by the concerns on their safety, robustness, and efficiency. In this work, we propose COCKTAIL, a novel design framework that automatically learns a neural network based controller from multiple existing control methods (experts) that could be either model-based or neural network based. In particular, COCKTAIL first performs reinforcement learning to learn an optimal system-level adaptive mixing strategy that incorporates the underlying experts with dynamically-assigned weights, and then conducts a teacher-student distillation with probabilistic adversarial training and regularization to synthesize a student neural network controller with improved control robustness (measured by a safe control rate metric with respect to adversarial attacks or measurement noises), control energy efficiency, and verifiability (measured by the computation time for verification). Experiments on three non-linear systems demonstrate significant advantages of our approach on these properties over various baseline methods.**

I. Introduction

Machine learning techniques, particularly those based on neural networks, have seen rapidly growing applications in autonomous cyber-physical systems such as self-driving vehicles, smart buildings, and robotic systems. These learning-enabled cyber-physical systems (LE-CPSs) adopt machine learning techniques not only for perception of the environment [1], but increasingly also for control [2] and decision making, in large part due to their advantages in learning effective strategies without the need of developing complex, costly, and error-prone physical models [3]. However, applying neural networks for building autonomous CPSs still faces significant hurdles, particularly with concerns of their impact on system safety, robustness, and efficiency. To enable their wider adoption, it is important to develop automated design methods and tools for analyzing these properties and optimizing the control design accordingly.

In this paper, we present COCKTAIL, a novel framework for learning an improved neural network controller from multiple existing control methods ("experts"). This is based on the observation that for many control applications, there are often multiple candidate experts available [4]. They could be based on well-established model-based approaches, such as model-predictive control (MPC) [5] or linear quadratic regulator (LQR) [6]. They could also be neural network based control methods that are trained through different algorithms, e.g,. via various reinforcement learning (RL) approaches with different rewards functions and hyper-parameters. In practice, it is also common for LE-CPSs to have multiple available controllers that are designed by different teams and/or for different objectives.

The multiple available controllers/experts, which may include both model-based and neural network-based ones, often perform differently and have different strengths with respect to the changing system state. Thus, the first step of our framework COCKTAIL is to learn a system-level **adaptive mixing** strategy that linearly combines the multiple available experts with dynamically-assigned weights for generating control input to the system. The weights are adapted based on the system state at each sampling period, to optimize system control robustness and control energy efficiency. Note that the robustness objective is defined as a safe control rate metric (i.e., how likely the system can remain safe from any initial state) under optimized adversarial attacks or random measurement noises to the system state. We formulate this adaptive mixing problem as a Markov Decision Process (MDP) with a reward function modeling robustness and efficiency.

While the adaptive mixing strategy can leverage the strengths from multiple experts and effectively improve the control robustness and energy efficiency, the mixed controller design could take significant resources (e.g, in storage) to implement and very importantly, be difficult to formally verify its properties such as safety and robustness. Thus, the second step of COCKTAIL conducts a teacher-student **robust distillation** to synthesize a single student neural network from the mixed controller design, using a novel probabilistic adversarial training and regularization technique with dual-objective regression focusing on both robustness and verifiability (measured by the computation time for verification). As we observed in experiments, this provides significant further improvement on all the properties we consider, including robustness, verifiability, and energy efficiency.

Related work: Our work is related to a rich literature on adaptive controller design. For instance, simplex architecture [7] proposes a switching logic between a baseline controller and an advanced controller to improve the control performance. Control adaptation based on switching among multiple controllers/experts has also been addressed in [8] with a rule-based approach, in [4], [9] with DRL approaches, and in [10] with finite-size weighted adaptation based on Q-learning. Different from these discrete adaptation approaches, we consider a continuous version of adaptive mixing, whose feasible adaptation space is a super-space of the ones in these previous approaches. We find that by expanding the adaptation space, our approach can significantly improve the safe control rate over the literature.

Our work also relates to the knowledge distillation paradigm [11], where a complex neural network is distilled into a compact neural network with similar or even better performance. Distillation from multiple experts, i.e., an ensemble of teachers, has been considered in works such as [12], [13]. In these approaches, the weight for each teacher in the ensemble is pre-determined and the sum of the weights is constrained to 1. In contrast, our approach dynamically adjusts the weights with RL, and does not put constraint on the weight sum to facilitate the implementation of the RL process. Moreover, our distillation is based on a novel dual-objective process

We gratefully acknowledge the support from NSF grants 1834701, 1839511, 1724341, 2038853, 2048075, 2008827, 2015568, 1934931, and ONR grant N00014-19-1-2496, Simons Institute (Theory of Reinforcement Learning), Amazon, J.P. Morgan, and Two Sigma.

with consideration of both robustness and verifiability.

In summary, our work makes the following contributions:

- We propose the COCKTAIL framework to leverage multiple existing control methods (experts) and learn a better single neural network controller from them, with consideration of control robustness, control energy efficiency, and verifiability.
- The COCKTAIL framework includes two novel components. The adaptive mixing step uses RL to learn a system-level strategy for dynamically assigning weights in incorporating experts, with global optimum convergence assurance. The robust distillation step conducts probabilistic adversarial training and regularization to synthesize a single neural network controller that further improves the mixed controller design.
- Experiments on three non-linear systems demonstrate that our approach can significantly improve robustness, energy efficiency, and verifiability over various baseline methods, including any single expert and a state-of-the-art switching adaptation method from the literature.

In the rest of the paper, Section II presents the problem formulation. Section III presents our COCKTAIL framework. Section IV shows the experimental results, and Section V concludes the paper.

II. Problem Formulation

We consider a discrete-time feedback system with its dynamics as

$$s(t+1) = f(s(t), u(t), \omega(t), \delta(t)), \ \forall t \geq 0 \qquad (1)$$

where $f : \mathbb{R}^{|s|} \times \mathbb{R}^{|u|} \times \mathbb{R}^{|\omega|} \times \mathbb{R}^{|\delta|} \rightarrow \mathbb{R}^{|s|}$ is a locally Lipschitz-continuous function [14]. $s(t) \in \mathbb{R}^{|s|}$ is the system state vector. X is defined as the *safe region*, and any state out of X is considered unsafe. $X_0 \subseteq X$ is the set of all possible initial system states. $u(t) \in U \in \mathbb{R}^{|u|}$ is the feedback control input to the system plant at each timestep t, where U is the bound for vector $u(t)$. $\omega(t) \in \Omega \in \mathbb{R}^{|\omega|}$ is a bounded external disturbance. $\delta(t) \in \Delta$ is a perturbation to the system state that could be caused by targeted/optimized adversarial attacks or random measurement noises. Note that X, X_0, U, Ω, and Δ are constrained by pre-defined functions, such as boxes.

The above system can be controlled with a feedback controller κ that is either model-based or model-free (e.g., those based on neural networks). At each timestep t, the controller κ reads the system state $s(t)$, and computes a control input as $u(t) = \kappa(s(t))$. The system then evolves to $s(t+1)$ according to its dynamics in Eq (1). Such process repeats and a trajectory φ based on the system initial state $s(0) \in X_0$ and the controller κ can be defined as

$$\varphi_{s(0),\kappa}(t+1) = f(\varphi_{s(0),\kappa}(t), \kappa(\varphi_{s(0),\kappa}(t)), \omega(t), \delta(t)) \qquad (2)$$

A trajectory is safe if every state it visits is within the safe region X. For a controller κ, we can define a *safe initial state set* $X^{'}$, which includes any initial state whose trajectory under κ is safe, i.e.,

$$X_\kappa^{'} = \{s \mid s \in X_0, \varphi_{s,\kappa}(t) \in X, \ \forall t \geq 0\}$$

We can then define a *safe control rate* metric for each controller κ to measure how large its safe initial state set $X_\kappa^{'}$ is, with respect to the set of all possible initial states X_0 (i.e., the ratio between the sizes of the two sets).

Based the above system model, we define three properties for a controller κ as follows.

Property 1: **Control robustness** for a controller κ is defined as its safe control rate S_r under optimized adversarial attacks or random measurement noises on the system state (captured by the state perturbation $\delta(t)$). Note that system *safety* may be considered as a special case of robustness with 0 state perturbation.

Property 2: **Control energy efficiency** [4] for a controller κ is defined as the average control energy cost e (over T control steps) of the various trajectories generated from the initial states in its safe initial state set $X_\kappa^{\prime}$, i.e.,

$$e = \mathbb{E}\left[\sum_{t=0}^{T-1} ||\kappa(\varphi_{s,\kappa}(t))||_1\right], \forall s \in X_\kappa^{'} \qquad (3)$$

where $\|\cdot\|_1$ is the 1-norm operator.

Property 3: **Verifiability** is measured by the computation time of the verification processes for various properties on a given platform.

The problem we try to solve is then defined as: given a system as described in Eq (1) and multiple control experts $\kappa_i (i = 1, \cdots, n)$(not necessary to be optimal), we will design a new neural network controller κ^* that optimizes control robustness, control energy efficiency, and verifiability.

III. Our Cocktail Framework

This section presents our proposed COCKTAIL framework for solving the above problem. As shown in Fig. 1, the COCKTAIL framework includes two novel components. First, a system-level adaptive mixing strategy linearly combines the multiple control experts for generating the control input to the system plant. The weights for the linear combination are dynamically adapted based on the system state, and learned via RL according to an MDP formulation that optimizes control robustness (i.e., safe control rate) and energy efficiency with global optimum assurance. Then, through teacher-student knowledge distillation, a student neural network κ^* is learned from the mixed controller design (which includes the underlying experts and the system-level neural network learned via RL for generating the weights). The distillation process is based on a probabilistic adversarial training and regulation technique that further improves control robustness and verifiability. Once we obtain the distilled student controller, formal verification is applied to analyze its safety. More details of COCKTAIL is shown in Algorithm 1 and introduced in the remaining of the section.

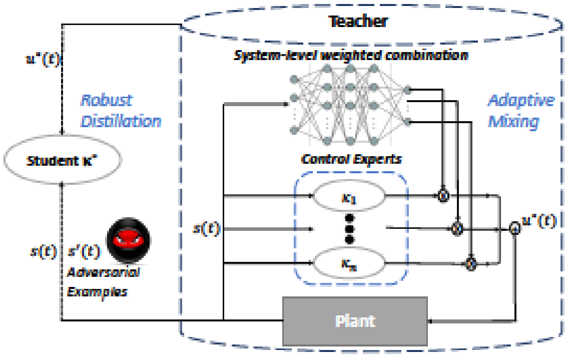

Fig. 1: Overview of the proposed COCKTAIL framework.

A. RL-based Adaptive Mixing of Multiple Experts

We propose to learn a system-level adaptive mixing strategy that significantly expands the action/adaptation space of the switching control methods in the literature (e.g., those in [4], [7]). In principle, we could build any mapping function $g : \mathbb{R}^{n \times |u|} \rightarrow \mathbb{R}^{|u|}$ that maps the various control input values computed by the experts to a control input for the system plant. In this work, we focus on linear mapping functions and dynamically adjust the weights for each expert based on the system state. To achieve this, we formulate the learning process for such adaptive mixing strategy as an MDP and solved with RL, with control robustness and energy efficiency as the reward.

Our MDP is captured with a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. $\mathcal{S}$ is the system state space, and $\mathcal{A}$ is the action space. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ describes the system dynamics. Constant $\gamma \in (0, 1]$ is discount factor. Parameterized by θ, policy $\pi_\theta \in \Pi : \mathcal{S} \rightarrow \mathcal{A}$ denotes the strategy. More specifically, they are formulated as follows.

Algorithm 1 Proposed COCKTAIL Framework

```
Input: Multiple control experts $\kappa_i, i = 1, \cdots, n$
Output: Student controller $\kappa^*(;q)$
Initialize replay memory $D$, adaptive policy network $\pi_\theta$, state perturbation bound $\Delta$, epochs $N$, steps $T$, Distillation epoch $N_E$, weights $\beta, \lambda$ and probability $p$.
for $epoch = 0, \ldots, N$ do
    Randomly initialize state $s(0) \in X_0$, $\theta_{old} \leftarrow \theta$.
    for $t = 0, \ldots, T$ do
        $a(t) = \pi_{\theta_{old}}(s(t))$.
        $u(t) = clip(\sum_{i=1}^{n} a(t)_i \times \kappa_i(s(t)), U_{inf}, U_{sup})$;
        $s(t+1) = f(s(t), u(t), \omega(t), \delta(t))$; Agent receives $r(t)$;
        $D.append([s(t), a(t), s(t+1), r(t)])$.

        /* RL(PPO) for adaptive mixing */
        Sample mini-batch from $D$; Compute advantage function $\hat{A}$

        $\theta = \arg\max_{\theta} \hat{\mathbb{E}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}\hat{A} - \beta\mathcal{KL}[\pi_{\theta_{old}}(\cdot|s), \pi_\theta(\cdot|s)]\right]$.
        /* Robust distillation */
        if $epoch \geq N_E$ then
            $z \xleftarrow[random]{uniform} [0, 1]$.

            $\delta = \Delta * sign(\nabla_s(l(\kappa^*(s;q), u)))$ if $z \leq p$ else 0.
            $q = \arg\min_{q} l(\kappa^*(s+\delta;q), u) + \lambda||q||_2^2$
        end
    end
end
```

State: $\mathcal{S}$ is the system state space. In this paper, we assume that each $s \in \mathcal{S}$ can be observed but may be maliciously attacked or affected by random measurement noises. The attacks or noises are captured by a bounded perturbation δ to the system state as introduced in Section II, and their effects reflect the control robustness.

Action: We consider a linear mapping function in this paper to generate the action space $\mathcal{A}$ for our adaptive mixing strategy. Specifically, at each timestep t, the action $a(t) = (a_1, \cdots, a_n)$ represents the weight assignment to the experts in the linear mapping function, where a_i is a bounded weight assigned to the i-th expert ($a_i \in [-A_{B_i}, A_{B_i}]$, $A_{B_i} \geq 1$). Then, the control input to the system is the weighted sum of the control inputs computed by the experts, with a clipping function ensuring its feasibility:

$$u(t) = clip(\sum_{i=1}^{n} a(t)_i \times \kappa_i(s(t)), U_{inf}, U_{sup}) \qquad (4)$$

where $\kappa_i(s(t))$ is the control input value computed by the i-th expert. U_{inf} and U_{sup} are the infimum and supremum of the control input vector bound U, respectively. Note that as a polyhedron, the action space in our approach is a super-space of the one in [10] (convex hull) and in [4], [9] (switching).

Reward function: The reward function encodes our desired goal for optimizing control robustness (i.e., safe control rate) and control energy efficiency, by steering the system away from the unsafe region and using as little energy as possible. Specifically, it is defined as

$$r(s, a) = \begin{cases} R_{pun}, & if\ s \notin X \\ h(||u||), & otherwise \end{cases}$$

where R_{pun} is a large negative punishment on safety violations (i.e., $s \notin X$). h is a monotonically decreasing function that computes energy consumption based on the control input $||u||$ in Eq (4).

With above design of the reward function, we formulate an optimization problem concerning robustness and efficiency as

$$\max J_{\pi_\theta} = \sum_{t=0}^{T-1} \mathbb{E}\left[\gamma^t \cdot r(s(t), a(t))\right]$$
$$s.t.\ s(t+1) = f(s(t), u(t), \omega(t), \delta(t)), s(0) \in X_0$$
$$a(t) = \pi_\theta(s(t))$$
$$-A_{B_i} \leq a(t)_i \leq A_{B_i}, \forall\ i = 1, \cdots, n$$

where T is an episodic control length.

For each iteration in the learning of the adaptive mixing strategy in Algorithm 1, we solve the above optimization problem with the gradient ascent towards the optimal weights for the experts, i.e.,

$$\theta = \arg\max_{\theta} \hat{\mathbb{E}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}\hat{A} - \beta\mathcal{KL}[\pi_{\theta_{old}}(\cdot|s), \pi_\theta(\cdot|s)]\right]$$

where $\hat{A}$ is the advantage function in RL, $\mathcal{KL}$ is the KL divergence, θ_{old} represents the parameters for the adaptive mixing policy network from the last iteration, and $\hat{\mathbb{E}}$ is an estimator (sample mean) for the expectation. Our approach can converge to the optimal weight assignment for the optimization problem, as explained below.

Proposition 1: Given multiple experts $\kappa_i(i = 1, \cdots, n)$, our RL-based approach can learn an optimal policy π^* for the adaptive weight assignment of experts, and outperform (or perform equally to) any single expert controller or any switching adaptation policy π_s.

Proof: First, according to [15], the actor-critic methods for proximal policy optimization (PPO) [16] with neural networks approximation converge to the global optimum at a sub-linear rate. This applies to our approach. Moreover, the action space of any switching adaptation policy that switches among controllers (e.g., the one in [4]) or of any policy with finite-size weighted adaptation (e.g., the one in [10]) is a sub-space of our action space. As global optimum is better than or equal to any local optimum, the optimal policy π^* obtained in our approach should outperform or perform equally to the ones from any single expert or switching policy.

Remark 1: The optimality assurance only applies to PPO in principle [15]. In practice, however, we find that other RL methods such as the deep deterministic policy gradient (DDPG) [17] can also achieve significant improvement.

B. Robust Distillation to a Single Neural Network Controller

The adaptive mixing strategy can effectively leverage the strengths from multiple experts to improve control robustness and energy efficiency. However, the learned mixed controller design, with the multiple experts and a neural network for the adaptive mixing policy, may consume significant resources in implementation. Moreover, it is hard to formally verify the properties for such mixed controller due to its complexity. This motivates us to synthesize a single and simpler neural network controller via knowledge distillation.

An important observation that drives our distillation is that for a neural network, both its verification complexity and its robustness are often affected by its Lipschitz constant L. Typically, the smaller the Lipschitz constant is, the more robust and more verifiable (e.g., taking less time to verify certain properties) the neural network is [18], [19].

Thus, the goals for our distillation of the student network are two folds: 1) to achieve similar control performance as the mixed controller design (i.e, the teacher), by minimizing a loss function that measures the regression error between the student and the teacher; and 2) to further improve system verifiability and control robustness via reducing the Lipschitz constant of the student network.

To achieve our dual objectives, we propose a hybrid probabilistic learning process by randomly selecting direct distillation or adversarial training with the fast-gradient sign method (FGSM) [20] and L-2 regularization to reduce L, as shown in Algorithm 1. Specifically, the part of the adversarial training with regulation solves a min-max problem each time as:

$$\min_{q}\big(\max_{||\delta||\leq\Delta} l(\kappa^{*}(s+\delta;q),u)+\lambda||q||_2^2\big)$$

where κ^{*} is the distilled student network with parameters q. δ bounded by Δ is the perturbation on the system state, which may be caused by adversarial attacks or measurement noises. l is the MSE loss function that measures the regression error between the student network and the teacher, and λ is the weight for the regularization. Intuitively, minimizing this training loss will regulate the local Lipschitz constant, as the output of neighbour region of s is expected to map closed to u. The inner max problem is solved by adversarial example generation with gradient ascent method and sign function as

$$\delta=\Delta * sign(\nabla_s(l(\kappa^{*}(s;q),u)))$$

Through this min-max optimization, the Lipschitz constant of the distilled student network can be significantly reduced, improving both system verifiability and control robustness.

C. Verification of the Neural Network based Controllers

Once we obtain the distilled student neural network κ^{*}, we may formally evaluate some of its properties such as safety and robustness, using techniques such as control invariant set computation and reachability analysis for safety verification. Intuitively, a control invariant set is a subset of the safe region that every possible trajectory starting from it will never leave it. To compute the invariant set, reachable analysis is used to compute the set (or an over-approximation of it) of all possible states the system may visit within a finite-horizon timestep. They are more formally defined as follows.

Definition 1: A control invariant set X_I is a subset of the safe region X that is defined as

$$X_I=\{s \mid \varphi_{s,\kappa}(t)\in X_I\in X,\ \forall t\geq 0,\ \forall\omega(t)\in\Omega\}$$

Note that any initial state within the invariant set is guaranteed to have infinite-time horizon safety as its possible trajectories are bounded within the invariant set.

Definition 2: The reachable set for an initial state $s(0)\in X_0$ is the set of states that the system may reach within T timesteps, i.e.,

$$X_R=\{\varphi_{s_0,\kappa}(t)\mid\forall\ s(0)\in X_0,\ \forall\ 0\leq t\leq T-1\}$$

Directly performing reachability analysis and safety verification on neural networks is intractable in most cases. Thus, we leverage the methods from [4], [21] by first over-approximating the neural network controller with a Bernstein polynomial under bounded errors (with partitioning technique [21] for reducing the approximation error), and then transforming the entire system (including the plant) into a hybrid system. The system safety and the robustness property (safe control rate under attacks or noises) can then be evaluated on the hybrid system with existing tools from [22], [23]. Specifically, in mathematical form, we first approximate the student network κ with a Bernstein polynomial as follows:

$$\kappa^{*}(x)\in B_d(x)+[-\epsilon,\epsilon],\ \forall x\in X$$

where d is the degree of the Bernstein polynomial and ϵ is the absolute approximation error bound. If the approximation error is too large, we can further partition the system state as:

$$\kappa^{*}(x)\in B_d^p(x)+[-\hat{\epsilon}^p,\hat{\epsilon}^p],\ \forall x\in X^p,\forall p=1,\cdots,P.$$

where P is the number of partitions and $\epsilon=\max(\hat{\epsilon}^p)$ is the approximation error. Such error will eventually be counted as an additional external disturbance into the original system as $\hat{\Omega}=\Omega\bigoplus\epsilon$, where $\bigoplus$ is the Minkowski summation operator.

Remark 2: Benefited from the robust distillation, the neural network controller κ^{*} generated by COCKTAIL with reduced Lipschitz constant is much more computationally efficient for verification purpose, compared with not only the mixed controller design (which is hard to verify with current tools) but also the student network generated from direct distillation (i.e., without adversarial training and regulation for reducing Lipschitz constant). This is due to the fact that larger Lipschitz constant leads to more sampling, more partitions, and higher order of Bernstein polynomials for approximating the neural network. Moreover, the transformed hybrid system also has more optimization variables and requires more resources to verify. Note, large Lipschitz constant of neural network controller is also expected to cause a significant impact on Verisig [18], [24].

However, while system safety under no attack or measure noise can be effectively verified for our test examples using the generated student neural network (and demonstrated in our experiments), accurately computing the control robustness under attacks and noises is still quite challenging with the current formal analysis techniques, as the over-approximation error cannot be effectively reduced within reasonable computation time in this case [21]. Thus, in our experiments, the safe control rate metric (i.e., robustness) for a controller is *estimated* by picking random samples from the initial state set X_0 and evaluating the system safety under the controller via simulations. This is also because the safety for some baselines methods cannot be formally analyzed in any case with the current tools.

IV. EXPERIMENTAL RESULTS

Test Systems: We conduct experiments on three non-linear systems: a Van der Pol's oscillator, a three-dimensional system from [25] (example 15), and a cartpole system. Each system has two available control experts κ_1 and κ_2, obtained by DDPG with different hyper-parameters, or in the case of the 3D system, DDPG and a model-based controller from [25]. More details are as follows.

1) The Van der Pol's oscillator is described as

$$\begin{cases}s_1(t{+}1)=s_1(t)+\tau s_2(t)\\ s_2(t{+}1)=s_2(t)+\tau[(1{-}s_1^2(t))s_2(t){-}s_1(t){+}u(t)]+\omega(t)\end{cases}\tag{5}$$

where $s(t)=(s_1(t),s_2(t))^{'}$ is the system state. $X=X_0=[-2,2]^2$ (for further control invariant analysis). $u(t)$ is the control input variable, and is bounded by $[-20,20]$. External disturbance ω is a random variable uniformly sampled from $[-0.05,0.05]$. $\tau=0.05$ is the sampling period. We assume that each control epoch consists of 100 control steps, i.e., $T=100$ in Eq (3).

2) The 3D system is defined as $\dot{x}=y+0.5z^2,\dot{y}=z,\dot{z}=u$, where system state $s=(x(t),y(t),z(t))^{'}$, $X=X_0=[-0.5,0.5]^3$,

Oscillator	κ_1	κ_2	A_S [4]	A_W	κ_D	κ^*
S_r (%)	85	79.4	88.4	98	98.4	**98.8**
e	94.1	97.9	94.2	96.3	94.6	**86.2**
L	35.4	15.1	-	-	20.5	**7.6**
3D system						
S_r (%)	91	88.6	96.8	98.2	97.6	**99**
e	16.6	16.6	13.5	12.7	12.3	**11.8**
L	251	**0.72**	-	-	12.1	7.1
Cartpole						
S_r (%)	81.6	84	90.4	**99**	**99**	98.6
e	106.1	74.7	84.8	28.8	29	**27.7**
L	359.7	303.9	-	-	126.1	**72.5**

TABLE I. Comparison of COCKTAIL with baselines. S_a is the safe control rate without attacks or measurement noises to the system state yet, e is the control energy consumption, and L is the Lipschitz constant. The baselines include κ_1 *only*, κ_2 *only*, switching adaptation method A_S, intermediate mixed controller A_W after adaptive mixing in COCKTAIL (no distillation), and direct distillation result κ_D from A_W (no consideration of robustness). κ_2 in the 3D system is a polynomial controller [25] and has a very small L. The Lipschitz constant for A_S and A_W cannot be measured and thus are denoted as '-'. We can see the significant improvement from our approach.

$u(t) \in U = [-10, 10]$, and $T = 100$. A sampling period $\tau = 0.05$ is used to discretize the ordinary differential equations (ODEs) into a discrete system.

3) The cartpole system is described as

$$\begin{cases} s_1(t+1) = s_1(t) + \tau s_2(t) \\ s_2(t+1) = s_2(t) + \tau s_{acc} \\ s_3(t+1) = s_3(t) + \tau s_4(t) \\ s_4(t+1) = s_4(t) + \tau \theta_{acc} \end{cases} \quad \begin{cases} \psi = \dfrac{u + m_p l s_4^2 \sin s_3}{m_t} \\ \theta_{acc} = \dfrac{(g \sin s_3 - \cos s_3 \psi) m_t}{l(1.333 - m_p (\cos s_3)^2)} \\ s_{acc} = \dfrac{\psi - m_p l \cos s_3 \theta_{acc}}{m_t} \end{cases}$$

with $m_c = 1, m_p = 0.1, m_t = 1.1, g = 9.8, l = 1, \tau = 0.02$, $T = 200$ and $s = (s_1, s_2, s_3, s_4)'$. $X = \{s| \ s_1 \in [-2.4, 2.4], s_3 \in [-0.209, 0.209]\}$ and $X_0 = [-0.2, 0.2]^4$ ($X_0 \subset X$ for further reachability analysis).

In our testing for each example, we randomly sample 500 initial system states from X_0, and compare the results from our COCKTAIL framework and other baselines. The comparison on control robustness and energy efficiency is based on simulations within a Python environment that we developed. The further analysis on verifiability, with safety consideration, is done via formal analysis as outlined in Section III-C.

Effectiveness of our approach over baselines: We compare the following methods to demonstrate the effectiveness of our approach: 1) using a single control expert, e.g., κ_1 *only* or κ_2 *only*; 2) a state-of-the-art switching adaptation control method from [4], denoted as A_S; 3) the intermediate mixed controlled design (i.e., before distillation) in COCKTAIL, denoted as A_W; 4) the direct distillation result from A_W without any adversarial training and regulation, denoted as κ_D; and 5) the robust distillation result from A_W, which is what our COCKTAIL eventually produces, denoted as κ^*.

The comparison results are shown in Table I. We can see that compared with κ_1, κ_2 and A_S (single expert or switching adaptation method), κ^* obtained from our COCKTAIL framework provides significant improvement on the safe control rate (without attacks or measurement noises to the system state yet) and control energy efficiency. Compared with the intermediate mixed controller design A_W and the direct distillation result κ_D, κ^* is easier to verify with the smaller Lipschitz constant (more about this later; note that the mixed controller design cannot be verified with current tools and does not have associated Lipshitz constant). Our approach also has smaller control energy consumption than A_W and κ_D.

	Under adversarial attacks		With measurement noises	
Oscillator	κ_D	κ^*	κ_D	κ^*
S_r (%)	95.2	**98.8**	98.4	**98.8**
e	837.3	**132.1**	383.8	**98.9**
3D system	κ_D	κ^*	κ_D	κ^*
S_r (%)	91.6	**98.2**	96	**98.8**
e	149.2	**25.7**	61.3	**15.5**
Cartpole	κ_D	κ^*	κ_D	κ^*
S_r (%)	92.2	**96**	96.4	**98.4**
e	30.6	**29.1**	31.1	**28.1**

TABLE II. Comparison of κ^* and κ_D under optimized adversarial attacks and measurement noises to the system state. κ^*(COCKTAIL) shows stronger robustness, indicating the efficacy of our robust distillation design. Note that while not shown in the table, A_W performs slightly worse than κ^* in energy efficiency, and other baselines perform much worse in both robustness and energy efficiency.

Further analysis on robustness and verifiability: We then further tested the effectiveness of our approach in improving control robustness and system verifiability, considering the cases where the system encounters adversarial attacks or measurement noises to the system state. Specifically, the measurement noise is a random variable sampled from an uniform distribution and added to the system state $s(t)$ at every step. The adversarial attack is generated by FGSM with a bound that is the same or larger than the one assumed in our robust distillation. In the experiments, the noises and the attacks are between $10\% - 15\%$ of the system state value bound. Table II shows the result comparison between our approach (generating κ^*) and direct distillation (generating κ_D). We can see that our approach benefits from the probabilistic adversarial training and robust distillation design, producing results that are more robust with respect to the adversarial attacks and measurement noises, as well as have smaller energy consumption. The control signal (and its energy consumption) from our results is also more stable under attacks, which is visualized in Fig. 2. Furthermore, we conducted formal analysis of the system properties (i.e., computing invariant set and conducting reachability analysis for safety verification) for the oscillator and the 3D system, respectively, as shown in Figs. 3 and 4. The results demonstrate the effectiveness of our COCKTAIL in reducing verification time.

V. CONCLUSION

In this paper, we propose a novel framework COCKTAIL to automatically learn an improved neural network controller from multiple control experts for LE-CPSs. Our approach first learns a system-level adaptive mixing strategy with optimal weights dynamically assigned to the experts using reinforcement learning, and then synthesize a single student neural network controller with robust distillation. Experiments demonstrate that our approach can significantly improve system control robustness, control energy efficiency, and verifiability.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016.

[2] L. Yang, F. Wan, H. Wang, X. Liu, Y. Liu, J. Pan, and C. Song, "Rigid-soft interactive learning for robust grasping," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1720–1727, 2020.

[3] S. Xu, Y. Wang, Y. Wang, Z. O'Neill, and Q. Zhu, "One for many: Transfer learning for building hvac control," *Buildsys*, 2020.

[4] Y. Wang, C. Huang, and Q. Zhu, "Energy-efficient control adaptation with safety guarantees for learning-enabled cyber-physical systems," in *International Conference on Computer-Aided Design (ICCAD)*, 2020.

[5] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control engineering practice*, 2003.

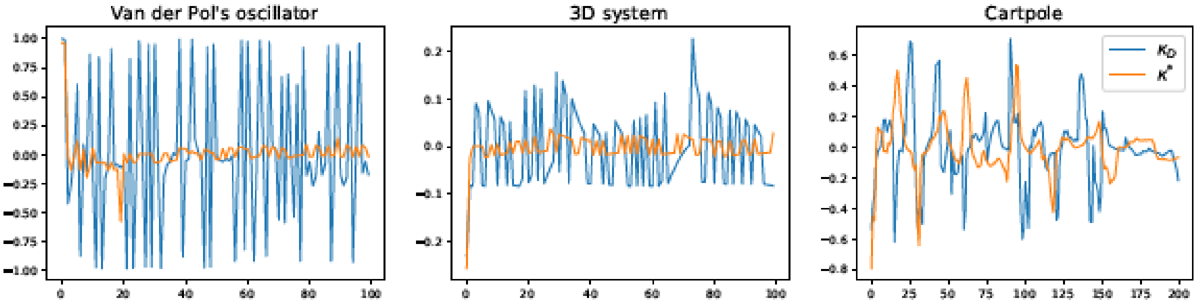

Fig. 2: The normalized control input signal $u(t)$ when the system encounters adversarial attacks. Compared with κ_D, κ^* obtained from COCKTAIL is more robust to these attacks and consumes much less energy, indicating the effectiveness of our robust distillation design. Note that the performance difference between κ^* and κ_D in cartpole is less significant than the others because cartpole is an unstable system.

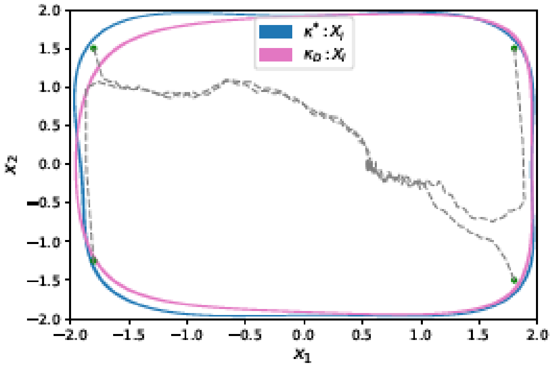

Fig. 3: Invariant set X_I of the oscillator system for κ^* and for κ_D. Although with higher order Bernstein polynomials, the X_I for κ_D is more conservative than the one for κ^* due to its slightly larger approximation error bound ϵ. X_I for κ^* is computed using the approach from [22] (as stated in Section III-C) in about 32 minutes, while needs around 11 hours for κ_D, showing the effectiveness of our approach in reducing verification time (improving verifiability). We also conducted 1500 simulations for different initial states within X_I for κ^*, and as expected, all the trajectories including the 4 dashed lines shown in the figure are indeed safe (the green dots are initial states; all 4 trajectories eventually are stable at around $(0.5, 0)$).

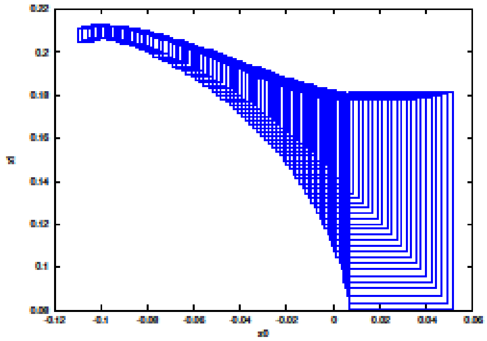

Fig. 4: Reachable set in the 3D system within the first 15 control steps from the initial state set on the upper left corner, as $s = (x, y, z) \in [-0.11, -0.105] \times [0.205, 0.21] \times [0.1, 0.11]$. Note that only (x, y) is plotted. We show this system because the difference between its Lipschitz constants of κ_D and κ^* is the smallest in all three examples. Nevertheless, κ_D cannot be verified because of a memory segmentation fault after 12 reachable set computation, caused by its large Lipschitz constant. In contrast for κ^*, the verification can be successfully completed within minutes (the reachable set does not go out of X and the system is verified to be $Safe$).

[6] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, 2002.

[7] D. Seto, B. Krogh, L. Sha, and A. Chutinan, "The simplex architecture for safe online control system upgrades," in *ACC*, vol. 6. IEEE, 1998.

[8] Z. Gong, J. I. Guzman, S. J. Scheding, D. C. Rye, G. Dissanayake, and H. Durrant-Whyte, "A heuristic rule-based switching and adaptive pid controller for a large autonomous tracked vehicle: from development to implementation," in *IEEE CCA*, 2004.

[9] C. Huang, S. Xu, Z. Wang, S. Lan, W. Li, and Q. Zhu, "Opportunistic intermittent control with safety guarantees for autonomous systems," *DAC*, 2020.

[10] S. Ramakrishna, C. Harstell, M. P. Burruss, G. Karsai, and A. Dubey, "Dynamic-weighted simplex strategy for learning enabled cyber physical systems," *Journal of Systems Architecture*, p. 101760, 2020.

[11] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[12] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran, "Efficient knowledge distillation from an ensemble of teachers." in *Interspeech*, 2017, pp. 3697–3701.

[13] Y. Chebotar and A. Waters, "Distilling knowledge from ensembles of neural networks for speech recognition." in *Interspeech*, 2016.

[14] W. Ruan, X. Huang, and M. Kwiatkowska, "Reachability analysis of deep neural networks with provable guarantees," *IJCAI*, 2018.

[15] B. Liu, Q. Cai, Z. Yang, and Z. Wang, "Neural trust region/proximal policy optimization attains globally optimal policy," in *NIPs*, 2019.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

[17] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning." in *ICLR*, 2016.

[18] J. Fan, C. Huang, W. Li, X. Chen, and Q. Zhu, "Towards verification-aware knowledge distillation for neural-network controlled systems," in *ICCAD*. IEEE, 2019, pp. 1–8.

[19] P. Pauli, A. Koch, J. Berberich, and F. Allgöwer, "Training robust neural networks using lipschitz bounds," *preprint arXiv:2005.02929*, 2020.

[20] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *ICLR*, 2015.

[21] C. Huang, J. Fan, W. Li, X. Chen, and Q. Zhu, "Reachnn: Reachability analysis of neural-network controlled systems," *TECS*, vol. 18, 2019.

[22] B. Xue and N. Zhan, "Robust invariant sets computation for switched discrete-time polynomial systems," *arXiv:1811.11454*, 2018.

[23] X. Chen, E. Ábrahám, and S. Sankaranarayanan, "Flow*: An analyzer for non-linear hybrid systems," in *CAV*. Springer, 2013, pp. 258–263.

[24] R. Ivanov, J. Weimer, R. Alur, G. J. Pappas, and I. Lee, "Verisig: verifying safety properties of hybrid systems with neural network controllers," in *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, 2019, pp. 169–178.

[25] M. A. B. Sassi, E. Bartocci, and S. Sankaranarayanan, "A linear programming-based iterative approach to stabilizing polynomial dynamics," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 462–10 469, 2017.