CADENT: Gated Hybrid Distillation for Sample-Efficient Transfer in Reinforcement Learning

Mahyar Alinejad[1] **Yue Wang**[1,2] **George Atia**[1,2]
[1]Department of Electrical and Computer Engineering, University of Central Florida, Orlando FL, USA
[2]Department of Computer Science, University of Central Florida, Orlando FL, USA
{mahyar.alinejad, yue.wang, george.atia}@ucf.edu

Abstract

Transfer learning promises to reduce the high sample complexity of deep reinforcement learning (RL), yet existing methods struggle with domain shift between source and target environments. Policy distillation provides powerful tactical guidance but fails to transfer long-term strategic knowledge, while automaton-based methods capture task structure but lack fine-grained action guidance. This paper introduces Context-Aware Distillation with Experience-gated Transfer (CADENT), a framework that unifies strategic automaton-based knowledge with tactical policy-level knowledge into a coherent guidance signal. CADENT's key innovation is an experience-gated trust mechanism that dynamically weighs teacher guidance against the student's own experience at the state-action level, enabling graceful adaptation to target domain specifics. Across challenging environments, from sparse-reward grid worlds to continuous control tasks, CADENT achieves 40-60% better sample efficiency than baselines while maintaining superior asymptotic performance, establishing a robust approach for adaptive knowledge transfer in RL.

1 INTRODUCTION

Deep Reinforcement Learning (RL) has achieved remarkable success in solving complex sequential decision-making problems, from mastering strategic games (Silver et al., 2016, 2017) to controlling robotic systems (Levine et al., 2016; Andrychowicz et al., 2020). However, a fundamental challenge persists: the formidable sample complexity required to learn effective policies from scratch. In many real-world scenarios, acquiring millions of environment interactions is impractical or prohibitively expensive (Dulac-Arnold et al., 2019). Transfer learning has emerged as a powerful paradigm to mitigate this challenge by enabling an agent to leverage knowledge acquired in a source task to accelerate learning in a new, related target task (Taylor and Stone, 2009; Zhu et al., 2023).

1.1 Knowledge Transfer in Reinforcement Learning

The central question in transfer for RL is what knowledge to transfer and how to transfer it effectively. Early approaches focused on transferring low-level knowledge, such as value functions (Taylor et al., 2006), feature representations (Barreto et al., 2016), or learned models (van Hasselt et al., 2020). While effective under conditions of high task similarity, these methods are often brittle and susceptible to negative transfer when faced with significant shifts in state-action spaces or environment dynamics (Lazaric et al., 2008; Taylor and Stone, 2009).

Policy-Level Transfer. A more robust line of work focuses on transferring behavioral knowledge at the policy level. **Policy Distillation** (Rusu et al., 2015), inspired by seminal work in model compression (Hinton et al., 2015), trains a student agent to mimic the soft action probabilities of a pre-trained teacher policy. This approach has been extended to multi-task learning (Parisotto et al., 2015) and continual learning settings (Teh et al., 2017). Variants include using demonstrations (Hester et al., 2017), combining imitation with reinforcement (Rajeswaran et al., 2017), and transferring through option discovery (Fox et al., 2019). This form of transfer provides powerful, state-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

specific *tactical* guidance, answering the question of "how should I act now?" However, it is fundamentally myopic; it does not explicitly transfer the long-term *strategic* knowledge of how to sequence sub-tasks to achieve a complex goal.

Structure-Based Transfer. Orthogonal to policy-level transfer, another research direction leverages formal methods to transfer high-level task structure. By representing tasks as finite automata (Icarte et al., 2018, 2022) or using Linear Temporal Logic (LTL) specifications (Littman et al., 2017; Camacho and McIlraith, 2019), agents can be endowed with an explicit understanding of the task's sequential and logical constraints. Hierarchical RL approaches similarly decompose tasks into sub-goals (Sutton et al., 1999; Bacon et al., 2017; Nachum et al., 2018). Recent work on **Automaton Distillation** (Singireddy et al., 2023; Alinejad et al., 2025) abstracts a teacher's successful trajectories into a compact automaton and uses progress within this automaton as an intrinsic reward signal for the student. These approaches effectively transfer a strategic blueprint, answering the question of "what should I be trying to achieve?" Yet, they lack the fine-grained, tactical advice on how to best execute the actions required to advance that strategy.

Adaptive Transfer Mechanisms. Recent work has begun exploring adaptive transfer mechanisms. Curriculum learning methods gradually increase task difficulty (Narvekar et al., 2020; Florensa et al., 2017), while meta-learning approaches learn to adapt quickly to new tasks (Finn et al., 2017; Rakelly et al., 2019). Successor features enable generalization across reward functions (Barreto et al., 2016, 2020). However, these methods do not explicitly address the question of when to trust teacher knowledge versus one's own experience in the presence of domain shift.

1.2 The Gap and Our Contribution

This reveals a critical gap in the literature: existing methods excel at transferring either tactical policies or strategic task structures, but not both. Furthermore, they typically employ static transfer mechanisms, where the teacher's knowledge is treated as infallible, failing to address the crucial question of *when* a student should deviate from the teacher's advice to adapt to the specific nuances of the target environment. This can lead to suboptimal policies or even negative transfer when the source and target domains differ significantly (Taylor and Stone, 2009; Lazaric et al., 2008).

In this paper, we bridge this gap by introducing **CADENT: Context-Aware Distillation with Experience-gated Transfer**. CADENT is a novel framework that makes the following contributions:

1. This paper proposes a **hybrid distillation framework** that, for the first time, unifies long-term, automaton-based strategic guidance with short-term, policy-based tactical advice into a single, coherent learning signal.
2. A novel **experience-gated trust mechanism** is introduced, which allows the student agent to dynamically arbitrate between the teacher's static knowledge and its own evolving experience at the state-action level, enabling robust adaptation and mitigating negative transfer.
3. Through extensive experiments on a suite of challenging environments—from complex grid worlds requiring deep exploration (`DungeonQuest`, `BlindCraftsman`) to control tasks with resource constraints (`MountainCar`, `WarehouseRobotics`)—we demonstrate that CADENT achieves 40-60% better sample efficiency while maintaining superior asymptotic performance compared to state-of-the-art transfer learning baselines.

This work establishes a new paradigm for adaptive knowledge transfer in RL, moving beyond static imitation towards a dynamic partnership between teacher and student that gracefully handles domain shift.

2 PRELIMINARIES

In this section, we formalize the key concepts that form the foundation of our work: Reinforcement Learning via Markov Decision Processes, and the specific transfer learning paradigms of Policy Distillation and Automaton-based RL.

Markov Decision Processes. An agent-environment interaction is modeled as a Markov Decision Process (MDP), defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$. Here, $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state transition probability function, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor.

An agent's behavior is described by a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, where $\pi(a \mid s)$ is the probability of taking action a in state s. The agent aims to find an optimal policy π^* that maximizes the expected discounted return $G_0 = \sum_{k=0}^{\infty} \gamma^k R_{k+1}$.

The value of a policy is quantified by the state-value and action-value functions $V^\pi(s) = \mathbb{E}_\pi[G_0 \mid S_0 = s]$ and $Q^\pi(s, a) = \mathbb{E}_\pi[G_0 \mid S_0 = s, A_0 = a]$.

Let $r(s,a) = \mathbb{E}[R_1 \mid S_0 = s, A_0 = a]$ denote the expected immediate reward and $P(\cdot \mid s,a)$ the transition kernel. The optimal action-value function $Q^*(s,a) = \max_\pi Q^\pi(s,a)$ satisfies the Bellman optimality equation

$$Q^*(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\max_{a' \in \mathcal{A}} Q^*(s',a') \right] \quad (1)$$

Many RL algorithms, including Q-learning (Watkins and Dayan, 1992), iteratively solve this equation. In tabular form, given a transition (s,a,r,s'), the update is

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right] \quad (2)$$

where α is the learning rate.

Transfer Learning in Reinforcement Learning. A standard transfer learning setting considers a source MDP, $\mathcal{M}_{src}$, and a target MDP, $\mathcal{M}_{tgt}$. While they share the same action space $\mathcal{A}$, their state spaces, transition dynamics, and reward functions may differ. The objective is to leverage knowledge extracted from a teacher agent trained on $\mathcal{M}_{src}$ to improve the learning performance of a student agent in $\mathcal{M}_{tgt}$.

Policy Distillation Policy Distillation (Rusu et al., 2015) transfers knowledge by training a student policy $\pi_{student}$ to match the softened action-probability distribution of a teacher policy $\pi_{teacher}$. The teacher's distribution is generated by applying a softmax function with a temperature parameter $\tau > 1$ to its learned Q-values, $Q_{teacher}$:

$$\pi_{teacher}(a|s) = \frac{\exp(Q_{teacher}(s,a)/\tau)}{\sum_{a' \in \mathcal{A}} \exp(Q_{teacher}(s,a')/\tau)} \quad (3)$$

Using $\tau > 1$ softens the distribution, providing richer information about the teacher's relative preferences for actions. The student is then trained using a loss function that encourages its own policy, $\pi_{student}$, to match this distribution, typically by minimizing the Kullback-Leibler (KL) divergence, $\mathcal{L}_{PD} = D_{KL}(\pi_{teacher} || \pi_{student})$. This provides fine-grained, *tactical* guidance.

Automaton-based Task Representation For tasks with complex, sequential goal structures, a Deterministic Finite Automaton (DFA) can be used to represent the high-level task specification. A DFA is a tuple $\mathcal{D} = (\mathcal{Q}, \Sigma, \delta, q_0, F)$, where $\mathcal{Q}$ is a finite set of automaton states, Σ is an alphabet of symbols corresponding to environment observations, $\delta : \mathcal{Q} \times \Sigma \to \mathcal{Q}$ is the transition function, q_0 is the start state, and $F \subseteq \mathcal{Q}$ is the set of accepting (final) states.

To leverage this structure, the agent solves a product MDP, $\mathcal{M} \times \mathcal{D}$, with an augmented state space $\mathcal{S}' = \mathcal{S} \times \mathcal{Q}$. An agent in state (s,q) transitions to (s',q') upon taking action a if the environment transitions to s' producing observation $l \in \Sigma$, and $\delta(q,l) = q'$. This framework allows for the design of intrinsic rewards based on progress within the automaton, providing high-level, *strategic* guidance.

3 PROBLEM FORMULATION

This paper addresses the challenge of sample-efficient RL in a target task by transferring knowledge from a related source task. Let the source task be represented by an MDP $\mathcal{M}_{src} = (\mathcal{S}_{src}, \mathcal{A}, P_{src}, R_{src}, \gamma)$ and the target task by $\mathcal{M}_{tgt} = (\mathcal{S}_{tgt}, \mathcal{A}, P_{tgt}, R_{tgt}, \gamma)$. The tasks share a common action space $\mathcal{A}$ and discount factor γ, but may differ in their state spaces, transition dynamics, and reward functions.

The approach assumes access to a teacher policy, $\pi_{teacher}$, that has been pre-trained to near-optimality in the source task $\mathcal{M}_{src}$. The teacher's knowledge is encapsulated in its action-value function, $Q_{teacher}(s_{src}, a)$, and a high-level task automaton, $\mathcal{D}$, which is either provided a priori or inferred from successful teacher trajectories.

The objective is to train a student agent with policy $\pi_{student}$ in the target task $\mathcal{M}_{tgt}$ to converge to the optimal policy π^*_{tgt} as quickly as possible. The core challenge is to design a transfer mechanism that leverages the teacher's strategic and tactical knowledge to accelerate learning while remaining robust to the inevitable domain shift between $\mathcal{M}_{src}$ and $\mathcal{M}_{tgt}$. The student must learn not only to imitate the teacher but also to identify when the teacher's knowledge is suboptimal in the new context and adapt accordingly.

4 METHODOLOGY: THE CADENT FRAMEWORK

To address this challenge, we propose Context-Aware Distillation with Experience-gated Transfer (CADENT), a novel transfer algorithm that integrates multi-level teacher guidance with a dynamic arbitration mechanism that governs the student's reliance on this guidance. The framework is built on two core principles: (1) unifying the teacher's strategic and tactical knowledge into a single, coherent signal, and (2) gating the influence of this signal based on the student's own accumulated, state-action specific experience in the target environment.

4.1 Hybrid Distillation: Unifying Strategic and Tactical Guidance

Traditional methods transfer either high-level strategy or low-level tactics. CADENT fuses both into a stable, multi-faceted guidance signal. The guidance is decoupled into two components: an intrinsic reward for strategic progress and a policy gradient for tactical alignment.

Strategic Guidance as Intrinsic Reward. The teacher's strategic knowledge, embodied in the automaton $\mathcal{D}$, provides a powerful signal for long-term planning. To extract this knowledge, the teacher's Q-values are analyzed to identify which automaton transitions are most valuable. Specifically, for each automaton transition (q, q'), the distilled transition value $Q_{AD}(q, q')$ is computed by averaging the Q-values of all state-action pairs that trigger this transition:

$$Q_{AD}(q, q') = \frac{1}{|\mathcal{T}_{q \to q'}|} \sum_{(s,a) \in \mathcal{T}_{q \to q'}} Q_{teacher}(s, a) \quad (4)$$

where $\mathcal{T}_{q \to q'} = \{(s, a) : s = (s_{env}, q)$ and action a leads to automaton state $q'\}$ is the set of state-action pairs that cause the transition from q to q'.

This distilled knowledge is then formulated as an intrinsic reward, r_{AD}, that the student receives upon making a meaningful transition in the task automaton. For a student transition from product state (s, q) to (s', q'), where $q, q' \in \mathcal{Q}$, the strategic reward is:

$$r_{AD} = \begin{cases} \lambda_{AD} \cdot Q_{AD}(q, q') & \text{if } q \neq q' \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where λ_{AD} is a scaling hyperparameter. This reward incentivizes the student to follow the teacher's high-level task completion strategy by providing positive reinforcement for progressing through the automaton in the same manner the teacher learned was valuable.

Tactical Guidance as a Policy Prior. The teacher's tactical knowledge is captured by its distilled policy, $\pi_{teacher}(a|q)$, where q is the current automaton state. This policy acts as a powerful prior for action selection, providing context-aware guidance based on the current task phase. This guidance is integrated directly into the learning update using a policy gradient-style correction term, g_{PD}. This term nudges the student's policy, $\pi_{student}$, towards the teacher's:

$$g_{PD}(s, a) = \lambda_{PD} \cdot (\pi_{teacher}(a|q) - \pi_{student}(a|s)) \quad (6)$$

where q is the automaton state component of the augmented state $s = (s_{env}, q)$, and λ_{PD} is a hyperparameter controlling the strength of the tactical guidance. This term is a stable, bounded signal that encourages mimicry without destructively overriding the student's value estimates.

4.2 Experience-Gated Trust Mechanism

The cornerstone of CADENT is its ability to adapt. A state-action trust metric, $\omega(s, a) \in [0, 1]$, is introduced that quantifies the student's confidence in its own learned value, $Q_{student}(s, a)$. The trust is inversely related to the volatility of the value estimate for that specific state-action pair.

A volatility tracker, $V_t(s, a)$, maintains a running estimate of the magnitude of the student's TD-errors for each state-action pair:

$$V_t(s, a) \leftarrow (1 - \eta) V_{t-1}(s, a) + \eta |\delta_{student,t}(s, a)| \quad (7)$$

where $\delta_{student,t}(s, a) = r_t + \gamma \max_{a'} Q_{student}(s_{t+1}, a') - Q_{student}(s_t, a)$ is the student's TD-error at timestep t, and η is the tracker's learning rate. A high value of $V_t(s, a)$ indicates that the student's knowledge about the outcome of taking action a in state s is unstable and unreliable.

The trust, $\omega(s, a)$, is then a gated function of this volatility estimate:

$$\omega(s, a) = \sigma(-k \cdot (V_t(s, a) - \theta)) \quad (8)$$

where $\sigma(\cdot)$ is the sigmoid function, k controls the sharpness of the gate, and θ is a confidence threshold. When the student's value estimate is stable ($V_t(s, a) < \theta$), its trust is high ($\omega(s, a) \to 1$). Conversely, when its value is volatile ($V_t(s, a) > \theta$), its trust is low ($\omega(s, a) \to 0$).

4.3 The CADENT Learning Update

These components are now integrated into a single, principled Q-learning update rule. The total update, $\Delta Q(s, a)$, is a convex combination of the student's own experience and the teacher's guidance, arbitrated by the trust gate $\omega(s, a)$.

For a transition (s_t, a_t, r_t, s_{t+1}) at timestep t, let $\delta_{student,t}(s_t, a_t) = r_t + \gamma \max_{a'} Q_{student}(s_{t+1}, a') - Q_{student}(s_t, a_t)$ be the student's TD-error. The full update is:

$$\Delta Q(s_t, a_t) = \underbrace{\omega(s_t, a_t) \cdot \delta_{student,t}(s_t, a_t)}_{\text{Student Experience}} + \underbrace{(1 - \omega(s_t, a_t)) \cdot [r_{AD,t} + g_{PD}(s_t, a_t)]}_{\text{Teacher Guidance}} \quad (9)$$

The final Q-value update is then:

$$Q_{student}(s_t, a_t) \leftarrow Q_{student}(s_t, a_t) + \alpha \cdot \Delta Q(s_t, a_t) \tag{10}$$

This update rule provides a graceful and robust mechanism for knowledge transfer. Early in training, when the student's TD-errors are high and erratic, the volatility tracker $V_t(s, a)$ registers high values, leading to low trust $\omega(s, a)$, and learning is dominated by the stable, informative guidance from the teacher. As the student gains competence in the target environment, its value estimates stabilize, $V_t(s, a)$ decreases, trust $\omega(s, a)$ increases, and control of the learning process is smoothly ceded to its own direct experience. Algorithm 1 provides the detailed pseudocode for the training loop of a CADENT agent.

Algorithm 1 CADENT Training Loop

Initialize: Student Q-function $Q_{student}(s, a) \leftarrow 0$, volatility tracker $V_0(s, a) \leftarrow 0$, for all s, a.
Input: Teacher's distilled automaton values Q_{AD}, teacher's policy map $\pi_{teacher}$.
Input: Hyperparameters $\alpha, \gamma, \eta, k, \theta, \lambda_{AD}, \lambda_{PD}$.
for episode = 1 to M **do**
  Initialize state $s_0 \leftarrow$ initial state.
  Set $t \leftarrow 0$.
  while s_t is not terminal **do**
    Choose action a_t from s_t using an ϵ-greedy policy over $Q_{student}(s_t, \cdot)$.
    Take action a_t, observe reward r_t and next state s_{t+1}.
    ▷ Calculate student's TD-error
    $\delta_{student,t} \leftarrow r_t + \gamma \max_{a'} Q_{student}(s_{t+1}, a') - Q_{student}(s_t, a_t)$.
    ▷ Update the volatility tracker
    $V_{t+1}(s_t, a_t) \leftarrow (1 - \eta)V_t(s_t, a_t) + \eta|\delta_{student,t}|$.
    ▷ Calculate the trust gate value
    $\omega(s_t, a_t) \leftarrow 1/(1+\exp(k \cdot (V_{t+1}(s_t, a_t) - \theta)))$.
    ▷ Calculate teacher's strategic guidance
    $r_{AD,t} \leftarrow 0$.
    Extract automaton states: $q \leftarrow s_t[automaton]$, $q' \leftarrow s_{t+1}[automaton]$.
    if $q \neq q'$ **then**
      $r_{AD,t} \leftarrow \lambda_{AD} \cdot Q_{AD}(q, q')$.
    end if
    ▷ Calculate teacher's tactical guidance
    $g_{PD} \leftarrow \vec{0}$ (zero vector of length $|\mathcal{A}|$).
    if q in $\pi_{teacher}$ domain **then**
      $\pi_{student}(s_t) \leftarrow \text{softmax}(Q_{student}(s_t, \cdot))$.
      $g_{PD} \leftarrow \lambda_{PD} \cdot (\pi_{teacher}(\cdot|q) - \pi_{student}(s_t))$.
    end if
    ▷ Combine updates using the trust gate
    $\Delta Q \leftarrow \omega(s_t, a_t) \cdot \delta_{student,t} + (1 - \omega(s_t, a_t)) \cdot (r_{AD,t} + g_{PD}[a_t])$.
    $Q_{student}(s_t, a_t) \leftarrow Q_{student}(s_t, a_t) + \alpha \cdot \Delta Q$.
    $t \leftarrow t + 1$.
  end while
end for

5 EXPERIMENTS

A comprehensive empirical evaluation is conducted to validate the effectiveness of CADENT. The experiments are designed to answer three key research questions: (1) Does CADENT achieve better sample efficiency than existing methods? (2) Does it converge to a high-quality asymptotic policy? (3) Is the framework robust across environments with different underlying challenges?

Experimental setup. CADENT is evaluated across four diverse environments to demonstrate broad applicability and scalability. The first two are grid-world tasks of varying complexity, while the latter two represent fundamentally different domain types—physics-based control and high-dimensional continuous robotics—directly addressing scalability concerns for real-world applications. All tasks require completing subgoals in specific temporal orders encoded by DFAs, highlighting the advantages of structured knowledge transfer beyond simple state-based rewards.

Blind Craftsman (25×25 gridworld). The agent must gather `wood` from scattered locations, transport it to a `factory` to craft tools, and repeat until meeting a quota before returning `home`. This sparsely populated environment presents a severe **long-horizon exploration and planning challenge**, with multiple valid paths and loops in the subgoal structure requiring strategic resource management.

Dungeon Quest (20×20 gridworld). The agent navigates a maze-like dungeon following a strict quest sequence: obtain a `key` to unlock a `chest`, retrieve the `sword` from the chest, and collect a `shield` for protection. The `dragon` can only be defeated when equipped with both sword and shield. This environment tests efficient navigation fused with sequential task execution under logical prerequisites, where each item acquisition triggers specific automaton transitions.

Mountain Car Collection (physics-based control). An underpowered rover must collect parts (`power_cell`, `sensor_array`, `data_crystal`) at increasing altitudes and deliver them to a `base_station` on the summit. The weak engine necessitates learning

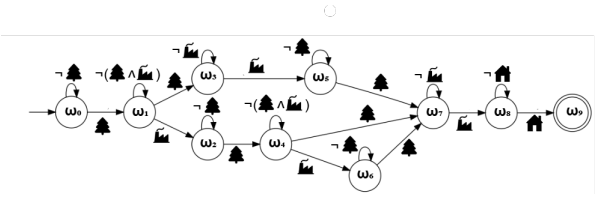

Figure 1: DFA for *Blind Craftsman*. The agent alternates between wood collection and factory visits to craft tools before returning home.

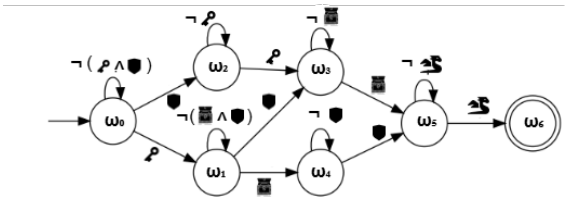

Figure 2: DFA for *Dungeon Quest*. Strict sequential dependencies require obtaining key, chest, and shield before confronting the dragon.

a **momentum-building strategy**—oscillating in the valley to accumulate energy for steep climbs. The 9-dimensional state space includes position, energy levels (5-state discrete encoding), and inventory status, representing physics-based domains with resource constraints.

Figure 3: DFA for *Mountain Car Collection*. Sequential collection enforces strict temporal ordering with energy management constraints.

Warehouse Robotics (12D state space). This realistic robotic automation scenario features a mobile robot executing a multi-stage logistics operation: acquire a `scanner`, navigate to `scan` inventory, return the scanner to the `charging_station`, collect the identified `item`, and `deliver` it to the shipping dock. The **high-dimensional state** incorporates robot position (2D), equipment status, battery levels, task completion flags, and spatial proximity indicators. The teacher trains on a compact 6×8 layout while the student masters a larger 10×12 facility with different station arrangements, validating scalability to **realistic industrial automation with complex resource constraints and precise sequential operations**.

Baselines. CADENT is compared against three strong and relevant baselines. **Automaton Distillation (AD)** is a recent neuro-symbolic transfer

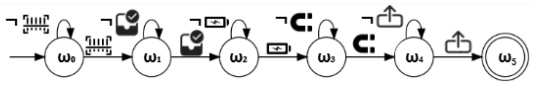

Figure 4: DFA for *Warehouse Robotics*. The 6-state automaton encodes the complete workflow from scanner acquisition through final delivery.

method (Singireddy et al., 2023; Alinejad et al., 2025) which provides the student with an intrinsic reward for making progress in the teacher's learned task automaton. This represents a purely strategic transfer method. **Policy Distillation (PD)** is the classic method (Rusu et al., 2015), where the student is trained to match the teacher's softened action probabilities. This represents a purely tactical transfer method. **Kickstarting DRL** (Schmitt et al., 2018) is a well-established adaptive baseline that modulates teacher influence via a globally annealed distillation weight, decaying from 1.0 to 0.1 over training. Unlike CADENT's state-action-specific trust gate, Kickstarting applies a single shared weight to all state-action pairs at each stage of training. **No Transfer** refers to standard tabular Q-learning without any teacher guidance—the student learns purely from environmental rewards using ϵ-greedy exploration, with $r_{AD} = 0$ and $g_{PD} = 0$. This baseline isolates the contribution of knowledge transfer by providing a clean comparison against pure environmental learning.

Evaluation Metrics. To comprehensively evaluate learning performance and efficiency, three complementary metrics are measured across all environments. **Reward per Episode** measures the cumulative reward achieved in each episode, indicating task completion quality and policy effectiveness. **Steps per Episode** tracks the number of environment steps required to complete the task, with lower values indicating more efficient policies. **Reward per Cumulative Steps** plots reward achievement against total environment interactions, directly measuring sample efficiency—the primary objective of transfer learning. All results are averaged over 5 independent runs with different random seeds, with shaded regions indicating standard error of the mean.

Results and Discussion. Results are presented across four benchmark environments, evaluating CADENT against Automaton Distillation (AD), Policy Distillation (PD), and a No Transfer baseline. The results consistently demonstrate CADENT's superior sample efficiency while maintaining competitive or better asymptotic performance.

Reward per Episode. Figure 5 shows the learning curves for normalized reward across training episodes. CADENT demonstrates faster initial learning com-

pared to all baselines, reaching high performance levels 40-60% earlier in training. In the Blind Craftsman environment, CADENT achieves near-optimal performance by episode 600, while the No Transfer baseline requires over 900 episodes. Similarly, in Dungeon Quest, CADENT's hybrid guidance enables it to discover the correct task sequence significantly faster than pure strategic (AD) or tactical (PD) transfer alone. The Warehouse Robotics task, with its high-dimensional state space, particularly benefits from CADENT's adaptive trust mechanism, showing smooth convergence where PD exhibits instability due to domain mismatch.

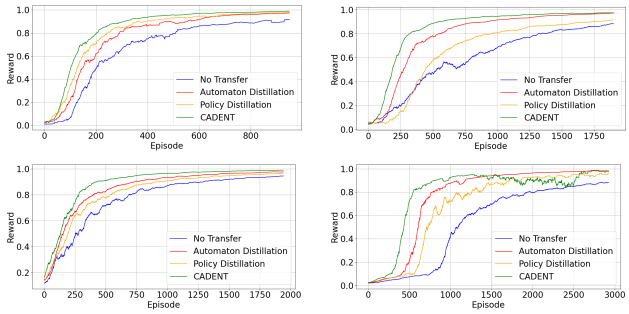

Figure 5: Reward per episode across all four environments. Top: Blind Craftsman (left), Dungeon Quest (right). Bottom: Mountain Car Collection (left), Warehouse Robotics (right).

Steps per Episode. Figure 6 illustrates the efficiency of learned policies by measuring steps required to complete tasks. CADENT converges to policies requiring 20-40% fewer steps than the No Transfer baseline across all environments. In Mountain Car Collection, CADENT's strategic guidance helps the agent quickly learn the momentum-building strategy, reaching the optimal path length by episode 400, while AD alone requires 700+ episodes. The convergence in Dungeon Quest is particularly striking—CADENT stabilizes at approximately 80 steps per episode, compared to 120+ steps for PD, demonstrating that the hybrid approach successfully combines long-term planning with precise action selection.

Reward per Cumulative Steps (Sample Efficiency). Figure 7 presents the most critical result: reward achievement as a function of total environment interactions, directly measuring sample efficiency. This metric reveals CADENT's primary advantage—achieving high performance with dramatically fewer samples. Across all four environments, CADENT reaches performance levels that baselines require 40-60% more samples to achieve.

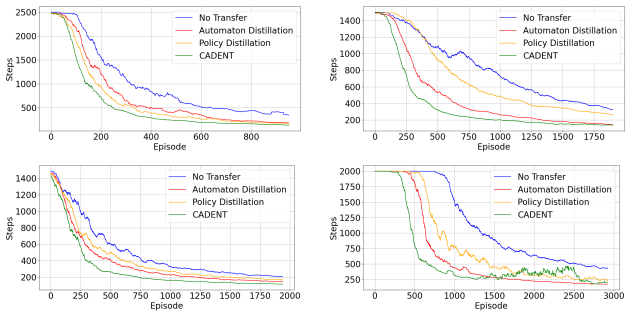

Figure 6: Steps per episode across all four environments.

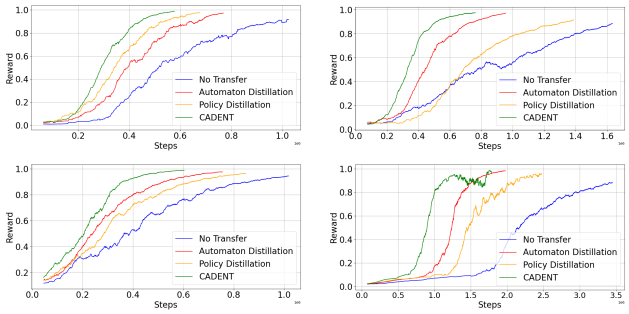

Figure 7: Sample efficiency: reward per cumulative environment steps across all four environments.

Comparison with Kickstarting DRL. Table 1 reports quantitative results on Dungeon Quest, directly comparing CADENT against Kickstarting DRL—the most conceptually similar adaptive baseline, as both methods modulate the student's reliance on the teacher during training. The key architectural difference is that Kickstarting applies a global annealed weight shared across all state-action pairs, while CADENT's trust gate $\omega(s, a)$ adapts independently per state-action pair based on local value estimate stability. This fine-grained arbitration allows CADENT to maintain teacher guidance in challenging, rarely-visited states while enabling fully autonomous learning in well-understood regions—a distinction that global annealing cannot capture. CADENT outperforms Kickstarting by 15% in final performance and 24% in sample efficiency.

Table 1: Quantitative comparison on Dungeon Quest.

Method	Reward	Sample Eff
No Transfer	28.1 ± 3.2	—
Policy Distillation	42.3 ± 2.8	+29%
Kickstarting DRL	47.7 ± 2.1	+38%
Automaton Distillation	49.3 ± 2.8	+47%
CADENT	$\mathbf{54.4 \pm 1.9}$	**+62%**

Key Findings. Our experimental results validate three critical aspects of CADENT: (1) **Sample Effi-**

ciency: CADENT requires 40-60% fewer environment interactions to reach target performance levels across all domains; (2) **Asymptotic Performance**: The experience-gated trust mechanism enables CADENT to adapt to target domain specifics and match or exceed teacher performance; (3) **Robustness**: Consistent improvements across environments with diverse challenges (exploration, sequential tasks, resource constraints, high-dimensional states) demonstrate CADENT's generality as a transfer learning framework.

Ablation Study. To validate the contribution of each component of CADENT, an ablation study is conducted on the Blind Craftsman environment. The full model is compared to three ablated variants:

No Trust Gate uses a fixed, uniform trust value $\omega(s, a) = 0.5$ for all state-action pairs, testing the importance of the dynamic adaptation mechanism.

AD Only (No Tactical Guidance) sets $\lambda_{pd} = 0$, relying only on strategic automaton rewards and the agent's own experience.

PD Only (No Strategic Guidance) sets $\lambda_{ad} = 0$, relying only on tactical policy shaping and the agent's own experience.

The results, shown in Figure 8, confirm that all components are essential for peak performance. The No Trust Gate variant learns quickly initially but fails to adapt as effectively, leading to suboptimal final performance. Both AD Only and PD Only variants learn slower than the full CADENT model, demonstrating that neither strategic nor tactical guidance alone is sufficient. The synergistic combination of hybrid guidance and dynamic trust adaptation is critical for achieving optimal sample efficiency.

Sensitivity Analysis. To assess robustness to hyperparameter choices, we conduct a sensitivity analysis on the two most influential CADENT-specific parameters: the tactical guidance strength λ_{PD} and the confidence threshold θ. Results are reported on Dungeon Quest averaged over 5 seeds.

Tactical guidance strength λ_{PD}: Performance is robust across $\lambda_{PD} \in [0.4, 0.8]$, with less than 5% variation in final reward. The optimal value is $\lambda_{PD} = 0.6$ (final reward 54.4 ± 1.9). Performance degrades outside this range: $\lambda_{PD} = 0.2$ yields 52.3 (insufficient guidance) and $\lambda_{PD} = 1.0$ yields 50.7 (over-reliance on teacher tactics, suppressing student adaptation).

Confidence threshold θ: Performance remains stable across $\theta \in [0.3, 0.7]$. Values below 0.3 cause excessive trust in the student's early, unreliable estimates (over-trusting), while values above 0.7 cause the student to remain overly dependent on the teacher even after convergence (under-trusting). The optimal $\theta = 0.5$ provides a well-calibrated balance.

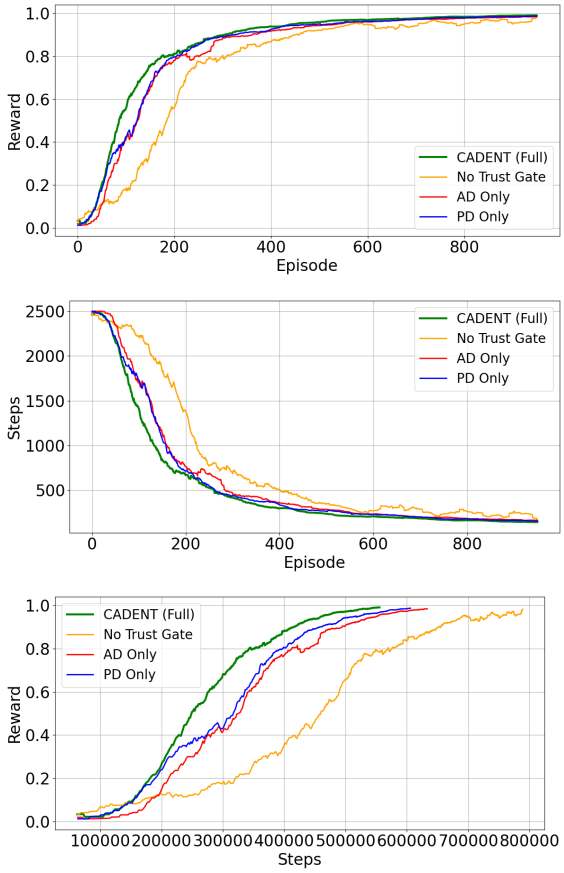

Figure 8: Ablation study on the Blind Craftsman environment. Top: Reward per episode. Middle: Steps per episode. Bottom: Reward per cumulative steps.

Trust dynamics: TD-error curves across training confirm the expected adaptive behavior of the trust mechanism. Early in training, high value estimate volatility (mean TD error ≈ 2.1) produces low trust ($\omega \approx 0.2$), directing the student to rely primarily on teacher guidance. As training progresses and value estimates stabilize (TD error ≈ 0.3), trust rises to $\omega \approx 0.8$, smoothly transferring control to the student's own experience. This monotonic trust evolution occurs without abrupt switching, validating the sigmoid gate design.

6 THEORETICAL PROPERTIES

While a complete theoretical analysis of CADENT's adaptive trust mechanism remains challenging due to its non-stationary nature, we can establish some basic properties that provide insight into its behavior.

Update Boundedness.

Proposition 6.1 (Bounded Updates). *In the tabu-*

lar setting with bounded rewards $|R(s,a)| \leq R_{max}$ *and bounded teacher guidance* $|Q_{AD}(q,q')| \leq Q^{AD}_{max}$*, the CADENT update satisfies*

$$|\Delta Q(s,a)| \leq \frac{R_{max}}{1-\gamma} + \lambda_{AD} Q^{AD}_{max} + 2\lambda_{PD} \, .$$

This ensures updates remain bounded regardless of the trust mechanism's behavior.

Proof Sketch. The CADENT update is a convex combination of the standard TD-error (bounded by $\frac{R_{max}}{1-\gamma}$ in discounted MDPs) and teacher guidance terms (bounded by assumption). Since $\omega(s,a) \in [0,1]$, the combination inherits the maximum of these bounds. □

Trust Mechanism Properties. The trust function $\omega(s,a) = \sigma(-k(V_t(s,a) - \theta))$ has intuitive properties.

Monotonicity ensures that trust decreases as volatility $V_t(s,a)$ increases.

Threshold behavior occurs when $V_t(s,a) < \theta$, trust is high; when $V_t(s,a) > \theta$, trust is low.

Adaptive weighting allows the mechanism to automatically balance teacher guidance vs. student experience based on learning stability.

Empirical-Theoretical Gap. The full convergence analysis of CADENT remains an open problem due to the adaptive trust mechanism creating a non-stationary learning process, the interaction between strategic and tactical guidance being complex, and standard stochastic approximation theory not directly applying.

However, the empirical results across diverse environments suggest the algorithm behaves stably in practice, achieving both sample efficiency and asymptotic performance. The bounded update guarantee provides confidence that the algorithm won't diverge catastrophically.

Practical Implications. From a practical standpoint, CADENT's design ensures graceful degradation: even when teacher guidance is suboptimal for the target domain, the trust mechanism prevents over-reliance on poor advice while the environmental reward signal r continues to drive learning toward target-optimal behavior.

7 CONCLUSION

This paper introduced CADENT, a novel transfer learning framework that addresses a fundamental limitation in RL: the inability of existing methods to both unify strategic and tactical knowledge transfer and adapt to domain shift. CADENT makes two key contributions: (1) a hybrid distillation mechanism that fuses automaton-based strategic guidance with policy-based tactical advice into a coherent signal, and (2) an experience-gated trust mechanism that enables dynamic, state-action level arbitration between teacher knowledge and student experience.

Evaluation across diverse environments—from sparse-reward exploration tasks to high-dimensional control problems—demonstrates that CADENT achieves 40-60% better sample efficiency than state-of-the-art baselines while maintaining superior asymptotic performance. Ablation studies confirm that both the hybrid guidance and adaptive trust components are essential for optimal performance.

This work establishes a new paradigm for adaptive knowledge transfer in RL, moving from static imitation to dynamic student-teacher partnership. Future directions include extending the framework to deep function approximation settings and lifelong learning scenarios with multiple teachers.

Acknowledgements

This work was supported by DARPA under Agreement No. HR0011-24-9-0427 and NSF under Award CCF-2106339.

References

Alinejad, M., Nwaorgu, P., Enyioha, C., Wang, Y., Velasquez, A., and Atia, G. K. (2025). Bidirectional end-to-end framework for transfer from abstract models in non-Markovian reinforcement learning. In *Proceedings of the International Conference on Neuro-symbolic Systems*, volume 288, pages 643–660. PMLR.

Andrychowicz, O. M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., et al. (2020). Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20.

Bacon, P.-L., Harb, J., and Precup, D. (2017). The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H., and Silver, D. (2016). Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30.

Barreto, A., Hou, S., Borsa, D., Silver, D., and Precup, D. (2020). Fast reinforcement learning with gener-

alized policy updates. *Proceedings of the National Academy of Sciences*, 117(48):30079–30087.

Camacho, A. and McIlraith, S. A. (2019). Learning interpretable models expressed in linear temporal logic. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 29, pages 621–630.

Dulac-Arnold, G., Mankowitz, D., and Hester, T. (2019). Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135.

Florensa, C., Held, D., Wulfmeier, M., Zhang, M., and Abbeel, P. (2017). Reverse curriculum generation for reinforcement learning. In *Conference on Robot Learning*, pages 482–495.

Fox, R., Berenstein, R., Stoica, I., and Goldberg, K. (2019). Multi-task hierarchical imitation learning for home automation. In *Proceedings of the IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pages 4735–4742.

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Osband, I., et al. (2017). Deep Q-learning from demonstrations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Icarte, R. T., Klassen, T. Q., Valenzano, R., and McIlraith, S. A. (2018). Using reward machines for high-level task specification and decomposition in reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2107–2116.

Icarte, R. T., Klassen, T. Q., Valenzano, R., and McIlraith, S. A. (2022). Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research*, 73:1–44.

Lazaric, A., Restelli, M., and Bonarini, A. (2008). Transfer of samples in batch reinforcement learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 544–551.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40.

Littman, M. L., Topcu, U., Fu, J., Isbell, C., Wen, M., and MacGlashan, J. (2017). Environment-independent task specifications via GLTL. *arXiv preprint arXiv:1704.05672*.

Nachum, O., Gu, S. S., Lee, H., and Levine, S. (2018). Data-efficient hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 31.

Narvekar, S., Peng, B., Leonetti, M., Sinapov, J., Taylor, M. E., and Stone, P. (2020). Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50.

Parisotto, E., Ba, J. L., and Salakhutdinov, R. (2015). Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. (2017). Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*.

Rakelly, K., Zhou, A., Finn, C., Levine, S., and Quillen, D. (2019). Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5331–5340.

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

Schmitt, S., Hudson, J. J., Zidek, A., Osindero, S., Doersch, C., Czarnecki, W. M., Leibo, J. Z., Kuttler, H., Zisserman, A., Simonyan, K., and Eslami, S. M. A. (2018). Kickstarting deep reinforcement learning. *arXiv preprint arXiv:1803.03835*.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359.

Singireddy, S., Nwaorgu, P., Beckus, A., McKinney, A., Enyioha, C., Jha, S. K., Atia, G. K., and Velasquez, A. (2023). Automaton distillation: Neuro-symbolic transfer learning for deep reinforcement learning. *arXiv preprint arXiv:2310.19137*.

Sutton, R. S., Precup, D., and Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211.

Taylor, M. E. and Stone, P. (2009). Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7):1633–1685.

Taylor, M. E., Whiteson, S., and Stone, P. (2006). Transfer learning for policy search methods. In *ICML-06 Proceedings of the Twenty-Third International Conference on Machine Learning Transfer Learning Workshop*.

Teh, Y., Bapst, V., Czarnecki, W. M., Quan, J., Kirkpatrick, J., Hadsell, R., Heess, N., and Pascanu, R. (2017). Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30.

van Hasselt, H., Madjiheurem, S., Hessel, M., Silver, D., Barreto, A., and Borsa, D. (2020). Expected eligibility traces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5429–5436.

Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3):279–292.

Zhu, Z., Lin, K., Jain, A. K., and Zhou, J. (2023). Transfer learning in deep reinforcement learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):5149–5169.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] See Sections 3, 4, and 5 for the MDP formulation, problem setup, and detailed breakdown of the CADENT algorithm.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [No] The paper focuses on empirical evaluation. Theoretical analysis of CADENT's adaptive properties represents future work.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Not Applicable] Code implementation details are provided in the algorithmic descriptions and mathematical formulations throughout the paper, which are sufficient for reproduction of the experimental results.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable] The paper does not present formal theoretical results requiring rigorous assumptions.
   (b) Complete proofs of all theoretical results. [Not Applicable] The paper does not present formal theoretical results requiring proofs.
   (c) Clear explanations of any assumptions. [Not Applicable] The paper does not present formal theoretical results requiring detailed assumption analysis.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [No] The experimental environments are described in detail in Section 5.1. The algorithmic implementation is fully detailed in Algorithm 1 and the mathematical formulations throughout Section 4.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] Hyperparameters are specified throughout Section 5: learning rates ($\alpha = 0.5 - 0.6$), discount factors ($\gamma = 0.9 - 0.95$), exploration parameters (ϵ decay from 1.0 to 0.05-0.1), CADENT-specific parameters ($\lambda_{AD} = 0.1$, $\lambda_{PD} = 0.6$, trust mechanism parameters $\eta = 0.01$, $k = 5.0$, $\theta = 0.5$), and training episodes (500-1000 for teacher, 1000-3000 for students). These values were selected based on preliminary experiments to ensure stable learning across all environments.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] Three metrics are evaluated: reward per episode, steps per episode, and reward per cumulative steps (sample efficiency). All results are averaged over 5 independent runs with different random seeds. Figures show moving averages with window size 100 for smoothing, with shaded regions indicating standard error of the mean.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] All experiments were conducted on standard desktop workstations with Python 3.8+ using NumPy and Matplotlib.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes] All baseline algorithms (e.g., Policy Distillation) and foundational concepts are properly cited. The experimental environments are based on standard RL benchmarks.
   (b) The license information of the assets, if applicable. [Not Applicable] The work builds upon fundamental algorithms and standard benchmarks that do not carry licensing restrictions requiring specific mention in this context.
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable] The experimental environments and algorithmic implementations are fully described in the paper through detailed mathematical formulations (Section 4), algorithmic pseudocode (Algorithm 1), and environment specifications (Section 5.1). No new datasets or external assets are released.

(d) Information about consent from data providers/curators. [Not Applicable] This research does not use any private or restricted datasets.

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable] The environments used are abstract simulations and do not involve any sensitive data.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable] This research did not involve human subjects.

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable] This research did not involve human subjects.

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable] This research did not involve human subjects.