RAIL: Representation-Aligned Imitation Learning for Student-Compatible Teacher Policies

Meraj Mammadov
Örebro University

Pedro Zuidberg Dos Martires
Örebro University

Johannes A. Stork
Örebro University

Abstract

Reinforcement learning (RL) from raw sensory inputs such as images or onboard sensors can be challenging due to high-dimensional observations and sparse rewards. A common strategy is to train a teacher policy with access to privileged state information and then distill it into a student policy that acts from raw inputs alone. However, when the teacher relies on information unavailable to the student, exact imitation may be impossible, creating an irreducible imitation gap. Existing approaches typically mitigate this mismatch through careful reward shaping or additional reinforcement learning on the student policy after the teacher has been trained. Instead, we introduce Representation-Aligned Imitation Learning (RAIL), which addresses the mismatch during teacher training itself. RAIL learns a latent representation shared across teacher and student observations using contrastive learning, and trains the teacher policy directly in this space. Because the teacher policy is constrained to operate on this shared space, it learns behaviors that are reproducible from student observations. This substantially reduces the imitation gap while preserving task performance. Across multiple environments, RAIL outperforms strong baselines without reward modification or post-hoc student fine-tuning, and surpasses direct reinforcement learning from raw observations. The learned representations further enable zero-shot transfer to new tasks through teacher-only training. Project site: https://merajmammadov.com/rail/.

1 Introduction

Training reinforcement learning policies directly on the target observation space can be impractical due to high dimensional observation space, sparse rewards, or costly exploration. A common strategy is therefore to first train a teacher policy with RL using privileged state information, and then to distill its behavior into a student policy that operates with the actual deployment observations [2]. For example, in navigation tasks the teacher may observe a full map, obstacle locations, or target coordinates, while the student must act on onboard camera images or limited local sensing.

While this teacher–student paradigm can greatly simplify training, it also introduces a fundamental mismatch: when the teacher exploits privileged variables that are not recoverable from the student's observations, the resulting policy may be impossible for the student to reproduce. In that case, no policy conditioned only on student observations can exactly match the teacher's decisions, leading to an irreducible imitation gap [31].

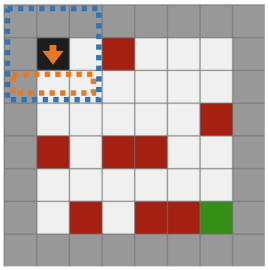


Figure 1: Observation mismatch in TunnelVision: Teacher (blue) vs. student (orange).

To illustrate this, consider the navigation task to the right (Figure 1), where an agent is dropped on the black tile and must reach the goal (green tile), while avoiding obstacles (red tiles). A privileged teacher observes all neighboring cells (dashed blue) and can move toward the goal without the need to change orientation. In contrast, the student observes only three cells in front of it (dashed orange) and needs to actively change its orientation to complete the mission successfully. This means that the

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

teacher can distinguish between states that are indistinguishable to the student, which in turn means that the teacher is capable of taking more informed actions than the student.

Existing approaches typically address this mismatch after teacher training, either by incorporating teacher signals into the student's RL objective [13, 14], or by additional RL fine-tuning of the student [33]. While effective in certain settings, these methods reintroduce the cost of RL optimization and do not remove the underlying information asymmetry.

In this work, we address the teacher–student imitation gap at its source: instead of training an unconstrained teacher and repairing the student afterward, we constrain the teacher to act only on information accessible to the student. We propose *Representation-Aligned Imitation Learning* (RAIL), which learns a common representation of teacher and student observations that captures only the information available to both. The teacher policy is trained with RL in this space, and the same policy is applied by the student at deployment time.

The contributions of this paper are: (1) We propose RAIL, a teacher–student imitation framework that enforces imitability during teacher training by restricting the teacher policy to a shared latent space that is also recoverable from student observations. (2) We introduce a contrastive representation-learning objective with gradient blocking, alignment, and stability regularization to prevent the leakage of private teacher information and enable direct policy transfer. (3) Across multiple environments, we show that RAIL reduces the imitation gap, outperforms strong baselines, learns faster than pixel-based RL, and supports zero-shot transfer to new tasks.

2 Preliminaries

We consider a sequential decision-making problem in which a teacher T and a student S act in the same environment but observe it through different observation mappings. Concretely, the environment is defined by the state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel $P(\cdot \mid s, a)$, and reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. At time-step t, both agents observe the same latent state $s_t \in \mathcal{S}$ through their corresponding observation mappings, $o_t^T = f_T(s_t)$ and $o_t^S = f_S(s_t)$, where $o_t^T \in \mathcal{O}_T$ and $o_t^S \in \mathcal{O}_S$ denote teacher and student observations, respectively. The teacher and student follow policies $a_t^T \sim \pi_T(\cdot \mid o_t^T)$ and $a_t^S \sim \pi_S(\cdot \mid o_t^S)$ that map their respective observations to a probability distribution over their shared action space.

In classical teacher–student settings [2, 14], the teacher policy π_T is trained with reinforcement learning to maximize expected discounted return:

$$\max_{\pi_T} \ \mathbb{E}_{\tau \sim \pi_T} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t^T) \right], \tag{1}$$

where $\tau = (s_0, a_0, \dots)$ is a trajectory generated by π_T under the transition kernel P and $\gamma \in (0, 1]$ is the discount factor. The student policy π_S is later trained to approximate the teacher's policy under its own observation space via imitation learning.

Given paired observations (o_t^T, o_t^S) from the same state s_t, standard behavior cloning [18] fits a student policy π_S to match a fixed teacher policy π_T by minimizing a divergence $D(\cdot, \cdot)$ between the action distribution of the teacher and student policies, conditioned on their respective aligned observations:

$$\min_{\pi_S} \ \mathbb{E}_{(o_t^T, o_t^S) \sim \mathcal{D}} \left[D\big(\pi_T(\cdot \mid o_t^T),\, \pi_S(\cdot \mid o_t^S)\big) \right], \tag{2}$$

where $\mathcal{D} = \{(o_t^T, o_t^S)\}_t$ is a dataset of aligned observations collected from trajectories $\tau \sim \pi_T$.

2.1 Irreducible imitation gap

Behavior cloning implicitly assumes that the teacher policy is realizable under the student's observation space. When this assumption holds, small imitation error translates to small performance gap between the agents. In particular, if the policy mismatch in Eq. (2) is upper-bounded by ε, and rewards are bounded as $|r(s, a)| \le R_{\max}$, the return gap satisfies the bound of Xu, Li, and Yu [34]:

$$V_{\pi_T} - V_{\pi_S} \le \frac{2\sqrt{2}\, R_{\max}}{(1-\gamma)^2} \sqrt{\varepsilon}, \tag{3}$$

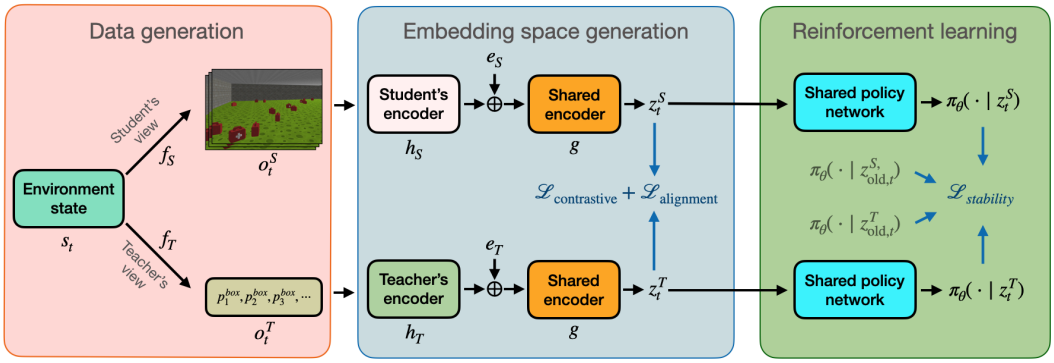


Figure 2: Left: teacher and student observations are generated from the same underlying state via different observation mappings. Middle: teacher and student encoders h_T and h_S, followed by a shared encoder g, map both observations into a common latent space; e_T and e_S are learned agent identity embeddings. Right: a shared policy π_θ operates on these representations to produce action distributions $\pi_\theta(\cdot \mid z_t^T)$ and $\pi_\theta(\cdot \mid z_t^S)$. $z_{\text{old},t}^T$ and $z_{\text{old},t}^S$ denote embeddings computed using a frozen copy of the encoders and are used to evaluate the stability objective $L_{\text{stability}}$ (cf. Subsection 3.3). Blue arrows indicate the losses used to train the encoders (cf. Section 3).

where V_π denotes expected discounted return. Thus, even small one-step imitation error can induce substantial long-horizon performance loss.

A sufficient condition for zero imitation gap is that the teacher action be conditionally independent of teacher observation given the student observation,

$$a^T \perp o^T \mid o^S. \tag{4}$$

That is, once o^S is known, any additional variation in the teacher's observation does not affect its action distribution. This implies that the teacher policy is a function of o^S alone and can therefore be matched exactly by a student policy conditioned on o^S.

In practice, however, the condition in Eq. (4) is often violated because the teacher bases its decisions on information unavailable to the student. Then there may exist two distinct states $s_1, s_2 \in \mathcal{S}$ that are identical to the student:

$$f_S(s_1) = f_S(s_2), \tag{5}$$

while the teacher distinguishes them and acts differently:

$$\pi_T(\cdot \mid f_T(s_1)) \neq \pi_T(\cdot \mid f_T(s_2)). \tag{6}$$

Because the student conditions only on $o^S = f_S(s_1) = f_S(s_2)$, it assigns the same action distribution to both states and therefore cannot match the teacher when distinguishing the states is necessary. The minimum achievable imitation loss is therefore strictly positive whenever such indistinguishable states occur with nonzero probability.

As shown by Ross, Gordon, and Bagnell [21], these local mismatches can compound over time, and we also show this in our experimental section (cf. Section 5). In particular, we show that such errors drive the student off-distribution, prevent recovery, and ultimately lead to large teacher–student performance gaps in state-of-the-art imitation learning approaches [18, 21], while RAIL successfully mitigates the presence of such scenarios and achieves a low imitation gap and high student performance.

3 Representation-Aligned Imitation Learning

RAIL enforces imitability at the level of the teacher policy by restricting it to depend only on information recoverable from the student observations. Instead of training an unconstrained teacher and compensating for the resulting mismatch afterward, we constrain the policy input during training so that all decisions are realizable under the student observation space.

RAIL builds on the view that teacher and student observations contain overlapping information about the same underlying state, together with private information available only to one side [4]. Exact

transfer requires the teacher policy to depend only on the overlapping information. We therefore learn encoders ϕ_T and ϕ_S that map teacher and student observations into a common latent space:

$$z_t^T = \phi_T(o_t^T), \qquad z_t^S = \phi_S(o_t^S). \tag{7}$$

For aligned observations from the same state s_t, we encourage z_t^T and z_t^S to agree while keeping representations of different states distinguishable. This naturally leads to a contrastive objective: paired teacher–student observations are treated as positives, while observations from different states act as negatives. Optimizing this objective encourages the latent space to retain information shared across both observation spaces while suppressing factors specific to each observation.

We train a single policy π_θ on top of this learned representation. During training, the teacher acts through teacher embeddings,

$$a \sim \pi_\theta(\cdot \mid z^T), \tag{8}$$

and at deployment the student applies the same policy to student embeddings,

$$a \sim \pi_\theta(\cdot \mid z^S). \tag{9}$$

Because the same policy π_θ operates on both teacher and student embeddings, transfer requires only replacing ϕ_T with ϕ_S once the embeddings are aligned.

An overview of the framework is shown in Figure 2. RAIL iteratively updates the encoder networks ϕ_T and ϕ_S with contrastive learning and the policy network π_θ with reinforcement learning on the learned embeddings. We describe this in more detail in Subsection 3.4.

To learn imitable teacher policies that allow direct transfer to the student, RAIL jointly optimizes three objectives: a contrastive loss to construct the shared latent space (cf. Subsection 3.1), an alignment loss to enable direct policy transfer (cf. Subsection 3.2), and a stability loss to regularize representation updates during policy learning (cf. Subsection 3.3).

3.1 Learning a shared latent space

We learn the shared latent space from aligned teacher–student observation pairs collected during teacher rollouts. For each pair (o_t^T, o_t^S) corresponding to the same state s_t, the embeddings z_t^T and z_t^S should agree, while embeddings from different states should remain distinguishable. We use a contrastive objective [15] as its positive-negative structure directly matches this goal.

We parameterize the encoders as $\phi_T = g \circ h_T$ and $\phi_S = g \circ h_S$, where h_T and h_S map teacher and student observations into feature spaces, and g maps both into a shared latent space. Given an observation o_i^A from agent $A \in \{T, S\}$, we compute

$$z_i^A = \phi_A(o_i^A) = \frac{g(h_A(o_i^A), e_A)}{\|g(h_A(o_i^A), e_A)\|_2}, \tag{10}$$

where $e_A \in \mathbb{R}^4$ is a learned indicator of the input source.

The contrastive loss over the embeddings is given by the symmetric InfoNCE objective [15]:

$$\mathcal{L}_{\text{contrastive}} = -\frac{1}{2N} \sum_{i=1}^{N} \left[\log \frac{\exp((z_i^T)^\top z_i^S / \tau)}{\sum_{j=1}^{N} \exp((z_i^T)^\top z_j^S / \tau)} + \log \frac{\exp((z_i^S)^\top z_i^T / \tau)}{\sum_{j=1}^{N} \exp((z_i^S)^\top z_j^T / \tau)} \right], \tag{11}$$

where z_i^T and z_i^S are unit-normalized embeddings of aligned observations, N is the batch size, and τ is a learnable temperature parameter.

For each embedding, the aligned embedding from the other agent is treated as positive, while all other embeddings from that agent in the batch serve as negatives. This encourages agreement only on information shared between the teacher and student observations and suppresses factors specific to each [8].

3.2 Aligning learned embeddings

The shared policy π_θ is trained on teacher embeddings but deployed on student embeddings. To enable this direct policy transfer, paired teacher–student observations must therefore map to nearly

identical latent embeddings. The contrastive objective (Eq. (11)) enforces relative similarity by distinguishing aligned pairs from negatives, but it does not minimize the distance between positive pairs directly. Therefore, small residual differences can remain and translate into an imitation gap, especially as reinforcement learning policies can be highly sensitive to input perturbations [5]. To address this, we introduce an absolute pairwise alignment objective that directly maximizes the cosine similarity between embeddings of paired observations:

$$\mathcal{L}_{\text{alignment}} = -\frac{1}{N}\sum_{i=1}^{N}\frac{(z_i^T)^\top z_i^S}{\|z_i^T\|_2\,\|z_i^S\|_2}. \tag{12}$$

Empirically, we find that this term is critical for reducing the imitation gap and improving transfer performance (cf. Section 5.3).

3.3 Stabilizing policy learning

In RAIL, the encoder parameters are updated throughout training, which induces a non-stationary input distribution for the policy. As the embedding space evolves, the same observation may map to different latent representations, resulting in unstable policy outputs. Moreover, contrastive representations may continue to drift even after alignment appears saturated [1] and further increase this non-stationarity.

We therefore regularize encoder updates to preserve the current policy behavior. Specifically, we penalize changes in the policy outputs induced by representation updates:

$$\mathcal{L}_{\text{stability}} = -\frac{1}{N}\sum_{i=1}^{N}\left(\frac{(u_{\text{old},i}^T)^\top u_i^T}{\|u_{\text{old},i}^T\|_2\,\|u_i^T\|_2} + \frac{(u_{\text{old},i}^S)^\top u_i^S}{\|u_{\text{old},i}^S\|_2\,\|u_i^S\|_2}\right), \tag{13}$$

where u_i^T, u_i^S denote the current outputs of the policy network (action logits for discrete policies and distribution parameters for continuous policies), and $u_{\text{old},i}^T, u_{\text{old},i}^S$ are the corresponding outputs for the same observations computed using a copy of the encoder frozen at the start of the encoder updates.

3.4 Training procedure

RAIL can be implemented as a standard reinforcement learning loop with an additional representation-learning phase inside each iteration. At iteration t, the agent first collects rollouts using the current teacher-side policy, $a \sim \pi_\theta(\cdot \mid \phi_T(o^T))$, exactly as in ordinary policy optimization [22]. The resulting rollout buffer contains rewards, actions, and paired teacher–student observations (o^T, o^S) from the same underlying states.

Before updating the policy, RAIL performs K encoder updates on minibatches sampled from this buffer. During this inner loop, the policy is held fixed and the encoders are trained to map paired teacher and student observations to a shared latent space. The encoder loss combines the contrastive, alignment, and stability objectives:

$$\mathcal{L}_{\text{embed}} = \lambda_{\text{con}}\mathcal{L}_{\text{contrastive}} + \lambda_{\text{align}}\mathcal{L}_{\text{alignment}} + \lambda_{\text{stab}}\mathcal{L}_{\text{stability}}. \tag{14}$$

The stability term is computed relative to a frozen copy of the encoders taken before the inner loop, so that representation updates preserve the current policy behavior. Note that in our experimental evaluation (cf. Section 5) we used $\lambda_{\text{con}} = \lambda_{\text{align}} = \lambda_{\text{stab}} = 1$ throughout.

After the encoder updates, the encoders are frozen and RAIL performs the usual RL policy update using teacher embeddings $\phi_T(o^T)$ from the rollout buffer. Crucially, gradients from the RL objective are not propagated into the encoders. Without this gradient blocking, policy optimization drives the representation to encode teacher-private information that improves teacher performance but is unrecoverable from student observations, thereby breaking direct transfer. Thus, RAIL keeps the outer loop identical to standard teacher-side RL while inserting an inner contrastive loop that maintains student-compatible representations. Algorithm 1 summarizes the overall training procedure.

One implementation detail not explicitly shown in the algorithm is the computation of the stability objective. At the beginning of each representation-learning phase, we store frozen copies of the current

Algorithm 1 Representation-Aligned Imitation Learning (RAIL)

Require: Teacher encoder ϕ_T, student encoder ϕ_S, policy π_θ

for each training iteration **do**
  Collect teacher rollouts using $\pi_\theta(\phi_T(o^T))$
  Store aligned teacher–student observations
  for K iterations with π_θ fixed **do**
    Update ϕ_T, ϕ_S with $\mathcal{L}_{\text{embed}}$
  end for
  Update policy π_θ with RL while keeping ϕ_T, ϕ_S fixed
end for

encoders, denoted $\bar{\phi}_T$ and $\bar{\phi}_S$. These reference encoders remain fixed throughout the subsequent K representation updates and define the policy behavior prior to representation refinement. The stability loss penalizes changes in the policy outputs induced by updates to the embeddings relative to these frozen reference encoders.

4 Related Work

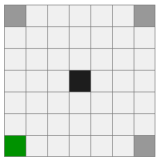
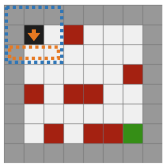
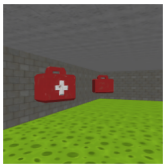
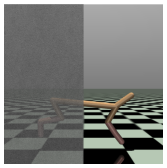
Teacher–student setups in imitation learning commonly train a privileged RL teacher with state information and use behavior cloning (BC) [18] or dataset aggregation (DAgger) [21] to distill it into a deployable student policy operating on restricted or high-dimensional observations [2, 35]. This improves teacher learning, but creates a fundamental mismatch: when the teacher uses information unavailable to the student, the student cannot, in general, reproduce its actions, which yields an irreducible imitation gap [25, 31, 32]. In this case, imitation marginalizes over information private to the teacher, leading to suboptimal student policies.

Most prior work treats this mismatch as a **student-side optimization problem** after the teacher has already been trained. Several methods follow imitation with additional reinforcement learning, either by bootstrapping the policy with imitation learning [33], incorporating teacher guidance into the student's RL objective [13, 31, 32], or adaptively querying the teacher to improve learning under partial observability [7]. COSIL [14] augments the student's RL objective with a soft imitation term that encourages agreement with a privileged teacher. These methods can recover performance when the student can compensate through exploration, but, as we show in Section 5, can severely hinder learning when the optimal behavior differs substantially across teacher and student observations.

A second line of work **enforces imitability during teacher training**. Monaci et al. [12] explicitly restrict the teacher observation space to state variables that are known to be recoverable by the student, which requires task-specific prior knowledge. Reward-shaping methods encourage student-compatible behavior through auxiliary objectives [11, 26], but involve careful reward design. SITT [10] jointly trains teacher and student and penalizes teacher behavior that the student cannot match. However, it modifies the original task objective and introduces a sensitive environment-dependent trade-off hyperparameter. RAIL instead preserves the original reward function and enforces imitability by training the teacher on a shared latent space that enables direct transfer of the learned policy.

More broadly, our work relates to representation learning methods for transfer under domain shifts. Shon et al. [24] construct a common latent space for robotic imitation of human motions. Subsequent work learns representations that are invariant to domain changes using multi-view contrastive objectives [4, 23] or adversarial domain confusion [27]. These approaches combine representation learning with policy training to remove nuisance variation (e.g., viewpoint or background), while the same task-related information is present across views. Under information asymmetry, however, transfer requires excluding information that is not recoverable from the student observations, which we enforce by stopping policy gradients from updating the encoders. Giammarino et al. [6] train adversarial imitation learning in a contrastive latent space and report failure under this constraint, which highlights the importance of our stability and alignment objectives.

Table 1: Environments and imitation gaps used in experiments. Further details and the reward functions are provided in Appendix A.

	 **Corners**	 **TunnelVision**	 **CollectHealth**	 **QuarterCheetah**
Task	Navigate from center to the hidden target corner (green).	Reach the goal (green) while avoiding obstacles (red).	Collect all health boxes in the room while avoiding walls.	Run forward as fast as possible.
Teacher obs.	Position, previous action, and the target corner identity.	Position, previous action, and occupancy of all neighboring cells (blue).	Position, previous action, and relative coordinates of all remaining boxes.	Full proprioceptive state (17D joint positions and velocities).
Student obs.	Position and previous action only; target identity is hidden.	Position, previous action, and three neighboring cells facing its current orientation (orange).	Stack of 5 limited field-of-view first-person RGB observations.	Third-person RGB observation with partial occlusion (left half is removed).
Action space	Move to any neighboring cells, $\|\mathcal{A}\| = 8$.	Move or rotate to compass directions, $\|\mathcal{A}\| = 8$.	Move in 8 directions or rotate $\pm 30°$, $\|\mathcal{A}\| = 10$.	Continuous torques, $\|\mathcal{A}\| = 6$.
Imitation gap	Teacher can move directly to the target, student must visit all corners.	Teacher can move directly to goal, student must turn to discover obstacles.	Teacher can move without turning, student must turn to discover next target.	Teacher has full state; student must infer dynamics and state from partial visual observations.

5 Experiments

We evaluate RAIL in simulation environments with asymmetric observation spaces that induce an irreducible imitation gap. The teacher has access to privileged information, whereas the student operates under restricted observations, which makes direct imitation of an unconstrained teacher insufficient. Table 1 summarizes the environments; full details are provided in Appendix A. Our experiments address four questions: (i) whether RAIL reduces the imitation gap, (ii) how it compares to strong baselines, (iii) the contribution of its components, and (iv) whether the learned representation can be used to zero-shot transfer new behaviors through teacher-only training.

5.1 Experimental setup

We compare RAIL against four imitation learning baselines, along with direct RL from student observations (**Student RL**) as a reference and which requires reward signals in the student domain.

(1) Behavior Cloning (BC) [18] We train the teacher with RL, and the student is trained offline to imitate teacher actions via supervised learning on collected trajectories.

(2) DAgger [21] We use the same state-based teacher as BC and iteratively collect transitions under the student policy, label them with teacher actions at the visited states, aggregate the dataset, and train the student via supervised imitation.

(3) COSIL [14] We train the student with RL augmented by a soft imitation term that encourages agreement with a privileged teacher. Following Messikommer et al. [10], we use the PPO variant with a fixed target divergence of 0.5.

(4) SITT [10] We jointly train the teacher with RL and student with BC and penalize the teacher when its actions are difficult to imitate. The trade-off parameter is selected via grid search.

We use Proximal Policy Optimization (PPO) [22] for all RL-based methods. All methods use comparable network capacity, matched training budgets, and identical PPO hyperparameters where applicable. Each method is trained with five random seeds and evaluated on 100 environment seeds per run; we report averages across seeds. Full implementation details are provided in Appendix C. The implementation is available at `https://github.com/meraccos/RAIL`.

Table 2: Teacher performance (T), student performance after transfer (S), and imitation gap ($\Delta = T - S$). Performance is measured by success rate for Corners, TunnelVision, and CollectHealth, and episodic return for QuarterCheetah. Success rates show the mean with upper and lower semideviations; QuarterCheetah returns show the mean $\pm$ standard deviation. Higher S and lower Δ are better.

	Corners			**TunnelVision**			**CollectHealth**			**QuarterCheetah**		
Method	T	$S \uparrow$	$\Delta \downarrow$	T	$S \uparrow$	$\Delta \downarrow$	T	$S \uparrow$	$\Delta \downarrow$	T	$S \uparrow$	$\Delta \downarrow$
BC	$1.00^{+0.00}_{-0.00}$	$0.24^{+0.03}_{-0.02}$	0.76	$1.00^{+0.00}_{-0.00}$	$0.12^{+0.03}_{-0.03}$	0.88	$1.00^{+0.00}_{-0.00}$	$0.28^{+0.11}_{-0.09}$	0.72	2186.83 $\pm$ 82.47	-268.17 $\pm$ 29.02	2455.00
DAgger	$1.00^{+0.00}_{-0.00}$	$0.38^{+0.15}_{-0.11}$	0.62	$1.00^{+0.00}_{-0.00}$	$0.15^{+0.02}_{-0.02}$	0.85	$1.00^{+0.00}_{-0.00}$	$0.84^{+0.02}_{-0.01}$	0.16	2186.83 $\pm$ 82.47	-484.25 $\pm$ 23.14	2671.08
COSIL	$1.00^{+0.00}_{-0.00}$	$0.28^{+0.09}_{-0.06}$	0.72	$1.00^{+0.00}_{-0.00}$	$0.00^{+0.00}_{-0.00}$	1.00	$1.00^{+0.00}_{-0.00}$	$0.08^{+0.01}_{-0.01}$	0.92	2186.83 $\pm$ 82.47	-308.04 $\pm$ 14.46	2494.87
SITT	$0.93^{+0.06}_{-0.13}$	$0.37^{+0.10}_{-0.11}$	0.56	$0.95^{+0.03}_{-0.03}$	$0.74^{+0.10}_{-0.16}$	0.21	$0.93^{+0.04}_{-0.04}$	$0.49^{+0.03}_{-0.02}$	0.44	962.67 $\pm$ 256.17	721.12 $\pm$ 239.63	241.55
RAIL (ours)	$1.00^{+0.00}_{-0.00}$	$\mathbf{1.00}^{+0.00}_{-0.00}$	**0.00**	$1.00^{+0.00}_{-0.00}$	$\mathbf{1.00}^{+0.00}_{-0.00}$	**0.00**	$0.99^{+0.00}_{-0.01}$	$\mathbf{0.98}^{+0.01}_{-0.02}$	**0.01**	2627.67 $\pm$ 33.83	**2510.24** $\pm$ 72.05	117.43
RAIL w/o stability	$1.00^{+0.00}_{-0.00}$	$1.00^{+0.00}_{-0.00}$	0.00	$0.55^{+0.29}_{-0.31}$	$0.53^{+0.16}_{-0.17}$	0.02	$0.97^{+0.02}_{-0.03}$	$0.95^{+0.04}_{-0.06}$	0.02	2352.79 $\pm$ 180.64	2343.32 $\pm$ 247.17	**9.47**
RAIL w/o alignment	$0.98^{+0.02}_{-0.04}$	$0.98^{+0.02}_{-0.04}$	0.00	$0.96^{+0.04}_{-0.07}$	$0.66^{+0.14}_{-0.22}$	0.30	$0.92^{+0.04}_{-0.06}$	$0.78^{+0.10}_{-0.12}$	0.14	2634.80 $\pm$ 75.38	2476.86 $\pm$ 116.82	157.94
RAIL end-to-end	$1.00^{+0.00}_{-0.00}$	$0.28^{+0.02}_{-0.02}$	0.72	$0.99^{+0.00}_{-0.00}$	$0.18^{+0.21}_{-0.11}$	0.81	$0.93^{+0.01}_{-0.01}$	$0.71^{+0.14}_{-0.11}$	0.22	2283.19 $\pm$ 91.35	1898.96 $\pm$ 569.92	384.23
RAIL sep. encoders	$0.91^{+0.07}_{-0.09}$	$0.89^{+0.08}_{-0.10}$	0.02	$0.95^{+0.04}_{-0.04}$	$0.60^{+0.09}_{-0.11}$	0.35	$0.98^{+0.01}_{-0.01}$	$0.98^{+0.01}_{-0.02}$	0.00	1813.80 $\pm$ 733.15	1690.38 $\pm$ 917.19	123.42
Student RL	–	$1.00^{+0.00}_{-0.00}$	–	–	$1.00^{+0.00}_{-0.00}$	–	–	$0.78^{+0.09}_{-0.08}$	–	–	1824.27 $\pm$ 30.35	–

5.2 Experimental results

We report teacher performance (T), student performance after transfer (S), and the imitation gap ($\Delta = T - S$). For Corners, TunnelVision, and CollectHealth, performance is measured by success rate. For QuarterCheetah, performance is measured by episodic return. Higher S indicates stronger transfer, while lower Δ indicates more reproducible teacher behavior. Results are summarized in Table 2, and evaluation curves for the higher-dimensional environments are shown in Figure 3.

Baseline results. BC performs poorly throughout. When the teacher relies on privileged information, identical student observations can correspond to contradictory teacher actions and BC cannot match these from student observations alone. DAgger improves over BC by mitigating covariate shift through student rollouts, but querying the same privileged teacher leaves this information mismatch unresolved. COSIL also performs poorly: its auxiliary imitation objective can conflict with reward maximization under student observations by encouraging actions that are suboptimal under student observations. SITT is the strongest prior baseline, indicating the benefit of changing the teacher's behavior to make it more imitable. However, reward-level regularization remains insufficient when the teacher's private information induces qualitatively different strategies. The student's gains also come with degraded teacher performance, which reflects an explicit trade-off between task optimality and imitability. Student RL can solve Corners and TunnelVision better than all other baselines, showing that these approaches can actually impede student performance.

To test whether the failure of the baselines arises from information mismatch, as we argue above, we repeat the training protocol but with matched information content: students receive teacher observations, or unmasked images in QuarterCheetah. In this experiment, all baselines match or approach teacher performance across environments, which supports our explanation (Appendix D.3).

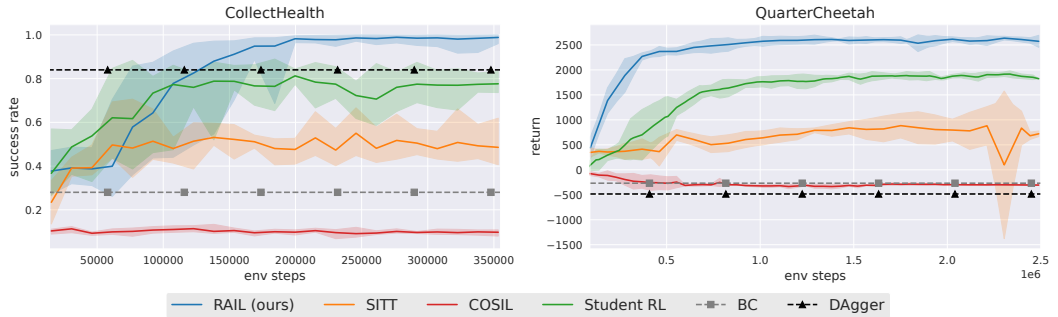


Figure 3: Student evaluation curves for CollectHealth and QuarterCheetah. CollectHealth reports success rate, while QuarterCheetah reports episodic return. BC and DAgger are shown as horizontal lines corresponding to their final student performance.

RAIL results. RAIL consistently achieves the highest student performance and the smallest imitation gap across all environments. In Corners and TunnelVision, it attains perfect transfer (S=1.00) with zero gap, fully matching teacher performance. In CollectHealth, it reaches S=0.98 with a negligible gap of 0.01 and substantially outperforms all baselines. Compared to Student RL, RAIL yields higher performance in CollectHealth (0.98 vs. 0.78) and QuarterCheetah (2510.24 vs. 1824.27), showing that improved representations and decoupled representation learning can translate into substantially better performance in high-dimensional observation spaces, which is consistent with prior findings in RL [9, 28]. Notably, RAIL maintains strong teacher performance while achieving strong transfer, unlike SITT, which reduces the gap at the cost of degraded teacher returns. This likely reflects the difficulty of balancing task reward and imitation constraints, together with sensitivity to the reward trade-off. Figure 3 further shows that RAIL reaches strong student performance faster than the online baselines in both high-dimensional environments.

Representation analysis. To assess what information can be recovered from the learned teacher embeddings, we train two-layer prediction networks on frozen embeddings in the Corners environment. Across ten seeds, prediction of the privileged target corner from the initial state achieves $25.4 \pm 1.4\%$ accuracy, close to the 25% chance level, whereas prediction of the agent's coordinates, which are available to both agents, achieves $100.0 \pm 0.0\%$. This provides evidence that the teacher embeddings retain shared state information while suppressing information private to the teacher.

5.3 Ablation study

The ablations in Table 2 (lower half) isolate the contribution of each component. Removing the alignment objective (**RAIL w/o alignment**) significantly reduces transfer performance, particularly in TunnelVision and CollectHealth. Although the contrastive objective encourages agreement, it does not enforce sufficient embedding similarity for direct policy transfer, and small mismatches in the embeddings translate into large imitation gaps.

Removing the stability objective (**RAIL w/o stability**) has little effect in Corners and CollectHealth, but reduces both teacher and student performance in TunnelVision and QuarterCheetah. The imitation gap nevertheless remains small across all environments, suggesting that stability primarily improves the teacher's training, while alignment helps keep student behavior close to the teacher's.

Allowing gradients from the policy to update the encoder (**RAIL end-to-end**) leads to large imitation gaps across all tasks despite strong teacher performance. This confirms that RL updates without gradient blocking drive the representation to encode teacher-specific information, resulting in policies that are no longer compatible with student embeddings.

Replacing the shared encoder with separate encoders (**RAIL sep. encoders**) reduces student performance and increases variability in Corners, TunnelVision, and QuarterCheetah, with little effect in CollectHealth. Consistent with prior work [17], this suggests that parameter sharing stabilizes the learning of common embeddings.

Finally, to assess RAIL's sensitivity to the contrastive batch size, we evaluate batch sizes from 128 to 1024 while holding the remaining training setup fixed. Performance remains high across this range, with larger batches typically improving success rates and reducing variability (Appendix D.2).

5.4 Zero-shot transfer to new tasks

We evaluate whether the learned representations support transferring new behaviors from the teacher agent without collecting additional student data. We modify Corners, TunnelVision and CollectHealth to support two tasks that require different control strategies while sharing the same observation manifold. The task changes involve switching from corner to side targets in Corners, adding diagonal actions in TunnelVision, and switching the rewarded object class in CollectHealth.

We first train RAIL on an initial task, freeze the encoder networks, and then train a new policy from scratch using only teacher observations on the second task. The resulting policy is transferred zero-shot by pairing it with the frozen student encoder. For comparison, we also train complete RAIL, including its encoders and policy, and Student RL from scratch on the second task.

Across all environments, policies trained on the frozen RAIL representation match or surpass the performance of Student RL, despite using no student observations during adaptation. This demon-

strates that the learned latent space captures rich environment structure shared across observations rather than features specific to a single objective. The transferred policies also perform on par with RAIL trained on the second task directly, indicating that the representation is sufficiently expressive to support new policies without modification. Detailed descriptions of the modified environments and complete experimental results are provided in Appendix B.

6 Conclusions

We introduced RAIL, a representation-level approach to teacher–student imitation under asymmetric observations. Rather than training a privileged teacher first and correcting the resulting mismatch afterward, RAIL constrains the teacher to act on representations that are also shared with the student. Empirically, RAIL achieves a near-zero imitation gap and outperforms strong baselines across environments. We further show that the learned representation can be reused for new tasks in the same environment through teacher-only policy training.

Limitations

RAIL assumes access to aligned teacher–student observations generated from the same underlying environment states, which is standard in teacher–student imitation learning but may not hold when observations are collected independently or only partially overlap temporally. Moreover, while our experiments show that gradient blocking and stability regularization substantially reduce the imitation gap, we do not provide formal guarantees that the learned policy is fully realizable under the student observation space. The zero-shot study evaluates task transfer within the same environments and observation spaces; the conclusions are therefore specific to this setting.

Acknowledgments and Disclosure of Funding

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The authors declare no competing interests.

References

[1] Roy Betser, Eyal Gofer, Meir Yossef Levi, and Guy Gilboa. “InfoNCE Induces Gaussian Distribution”. In: *International Conference on Learning Representations*. 2026.

[2] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. “Learning by Cheating”. In: *Conference on Robot Learning*. 2019.

[3] Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. *Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks*. 2023. arXiv: 2306.13831.

[4] André Correia and Luís A. Alexandre. “Multi-View Contrastive Learning from Demonstrations”. In: *IEEE International Conference on Robotic Computing*. 2022.

[5] Jesse Farebrother, Marlos C. Machado, and Michael Bowling. *Generalization and Regularization in DQN*. 2018. arXiv: 1810.00123.

[6] Vittorio Giammarino, James Queeney, and Ioannis Ch. Paschalidis. “Visually Robust Adversarial Imitation Learning from Videos with Contrastive Learning”. In: *International Conference on Robotics and Automation*. 2025.

[7] Yujin Kim, Nathaniel Chin, Arnav Vasudev, and Sanjiban Choudhury. *Distilling Realizable Students from Unrealizable Teachers*. 2025. arXiv: 2505.09546.

[8] Julius von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. “Self-Supervised Learning with Data Augmentations Provably Isolates Content from Style”. In: *Advances in Neural Information Processing Systems*. 2021.

[9] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. “CURL: Contrastive Unsupervised Representations for Reinforcement Learning”. In: *International Conference on Machine Learning*. 2020.

[10] Nico Messikommer, Jiaxu Xing, Elie Aljalbout, and Davide Scaramuzza. "Student-Informed Teacher Training". In: *International Conference on Learning Representations*. 2025.

[11] Nico Messikommer, Jiaxu Xing, Leonard Bauersfeld, Marco Cannici, Elie Aljalbout, and Davide Scaramuzza. *Approximate Imitation Learning for Event-based Quadrotor Flight in Cluttered Environments*. 2026. arXiv: 2603.07578.

[12] Gianluca Monaci, Michel Aractingi, and Tomi Silander. "DiPCAN: Distilling Privileged Information for Crowd-Aware Navigation". In: *Robotics: Science and Systems*. 2022.

[13] Malte Mosbach and Sven Behnke. "Grasp Anything: Combining Teacher-Augmented Policy Gradient Learning with Instance Segmentation to Grasp Arbitrary Objects". In: *IEEE International Conference on Robotics and Automation*. 2024.

[14] Hai Nguyen, Andrea Baisero, Dian Wang, Christopher Amato, and Robert Platt. "Leveraging Fully Observable Policies for Learning under Partial Observability". In: *Conference on Robot Learning*. 2022.

[15] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. *Representation Learning with Contrastive Predictive Coding*. 2018. arXiv: 1807.03748.

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems*. 2019.

[17] Petra Poklukar, Miguel Vasco, Hang Yin, Francisco S. Melo, Ana Paiva, and Danica Kragic. "Geometric Multimodal Contrastive Representation Learning". In: *International Conference on Machine Learning*. 2022.

[18] Dean A. Pomerleau. "ALVINN: An autonomous land vehicle in a neural network". In: *Advances in Neural Information Processing Systems*. 1989.

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. "Learning Transferable Visual Models From Natural Language Supervision". In: *International Conference on Machine Learning*. 2021.

[20] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. "Stable-Baselines3: Reliable Reinforcement Learning Implementations". In: *Journal of Machine Learning Research* 22.268 (2021).

[21] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning". In: *International Conference on Artificial Intelligence and Statistics*. 2011.

[22] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. *Proximal Policy Optimization Algorithms*. 2017. arXiv: 1707.06347.

[23] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. "Time-Contrastive Networks: Self-Supervised Learning from Video". In: *IEEE International Conference on Robotics and Automation*. 2018.

[24] Aaron P. Shon, Keith Grochow, Aaron Hertzmann, and Rajesh P. N. Rao. "Learning Shared Latent Structure for Image Synthesis and Robotic Imitation". In: *Advances in Neural Information Processing Systems*. 2005.

[25] Yuda Song, Dhruv Rohatgi, Aarti Singh, and J. Bagnell. "To Distill or Decide? Understanding the Algorithmic Trade-off in Partially Observable RL". In: *Advances in Neural Information Processing Systems*. Vol. 38, Main Conference. 2025. DOI: 10.52202/085713-2996.

[26] Yunlong Song, Kexin Shi, Robert Penicka, and Davide Scaramuzza. "Learning Perception-Aware Agile Flight in Cluttered Environments". In: *IEEE International Conference on Robotics and Automation*. 2023.

[27] Bradly C. Stadie, Pieter Abbeel, and Ilya Sutskever. "Third-Person Imitation Learning". In: *International Conference on Learning Representations*. 2017.

[28] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. "Decoupling Representation Learning from Reinforcement Learning". In: *International Conference on Machine Learning*. 2021.

[29] Emanuel Todorov, Tom Erez, and Yuval Tassa. "MuJoCo: A physics engine for model-based control". In: *International Conference on Intelligent Robots and Systems*. 2012.

[30] Mark Towers, Ariel Kwiatkowski, Jordan K. Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. "Gymnasium: A Standard Interface for Reinforcement Learning Environments". In: *Advances in Neural Information Processing Systems*. Vol. 38. 2025. DOI: 10.52202/085713-4916.

[31] Aaron Walsman, Muru Zhang, Sanjiban Choudhury, Dieter Fox, and Ali Farhadi. "Impossibly Good Experts and How to Follow Them". In: *International Conference on Learning Representations*. 2023.

[32] Luca Weihs, Unnat Jain, Iou-Jen Liu, Jordi Salvador, Svetlana Lazebnik, Aniruddha Kembhavi, and Alex Schwing. "Bridging the Imitation Gap by Adaptive Insubordination". In: *Advances in Neural Information Processing Systems*. 2021.

[33] Jiaxu Xing, Angel Romero, Leonard Bauersfeld, and Davide Scaramuzza. "Bootstrapping Reinforcement Learning with Imitation for Vision-Based Agile Flight". In: *Conference on Robot Learning*. 2024.

[34] Tian Xu, Ziniu Li, and Yang Yu. "Error Bounds of Imitating Policies and Environments". In: *Advances in Neural Information Processing Systems*. 2020.

[35] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. "End-to-End Urban Driving by Imitating a Reinforcement Learning Coach". In: *IEEE/CVF International Conference on Computer Vision*. 2021.

A Environments

Corners is a grid-world navigation task on an $n \times n$ grid (n=9), where the agent starts at the center and must reach a target corner sampled uniformly at the beginning of each episode. The action space consists of 8 discrete movements (cardinal and diagonal), with positions clipped to remain within the grid. Observations encode the agent position (x, y) as one-hot vectors of size n each, and the previous action as an 8-dimensional one-hot vector over $\{0, \ldots, 7\}$. The teacher additionally observes the target corner as a 4-dimensional one-hot vector, while the student does not, yielding $\dim(o^T) = 2n + 12$ and $\dim(o^S) = 2n + 8$. This private goal information induces an imitation gap, as the teacher can move directly to the target while the student must infer it through exploration. Episodes terminate upon reaching the target and are truncated after 60 steps. The reward function is $r_t = \mathbf{1}\{\text{goal reached}\} - 0.01$.

TunnelVision is a grid-world navigation task on an $n \times n$ grid (n=8) surrounded by a one-cell obstacle border, where the agent starts at $(1, 1)$ and must reach the fixed goal at $(n - 2, n - 2)$. Obstacles are sampled randomly at reset with probability 0.5, subject to constraints that remove diagonal obstacle patterns and preserve a valid path from start to goal. The agent can move or rotate in the four compass directions, yielding 8 discrete actions. Observations contain the local 3×3 occupancy grid, 4-dimensional one-hot-coded orientation, normalized previous action, and normalized position. The teacher observes all 9 cells in the local neighborhood, while the student observes only the 3 cells in front of its current orientation, with unseen cells masked by a constant; both observations have dimension $\dim(o^T) = \dim(o^S) = 16$. This asymmetry in information induces an imitation gap, as the teacher can solve the task without turning while the student must reorient to reveal hidden obstacles. Episodes terminate when the agent reaches the goal or collides with an obstacle/border and are truncated after 80 steps. The reward function is $r_t = 1 \times \mathbf{1}\{\text{goal reached}\} - 1 \times \mathbf{1}\{\text{collision}\} - 0.01 \times \mathbf{1}\{d_{t+1} \geq d_t\}$, where d_t is the Euclidean distance to the goal before the action.

CollectHealth is an episodic 3D navigation task based on MiniWorld [3], where the agent operates in a 15 m $\times$ 15 m room and must collect 5 randomly placed health boxes. The action space has 10 discrete actions: movement in 8 uniformly spaced relative directions with step size 0.5 m, and two camera rotations of $\pm 30°$. The teacher observes the relative positions of all remaining boxes in the agent frame, sorted by distance and padded with -1, together with the normalized agent position and previous action, giving $\dim(o^T) = 13$. The student observes a stack of 5 RGB first-person images, each resized to 84×84 and normalized to $[-1, 1]$, giving $\dim(o^S) = 5 \times 3 \times 84 \times 84$. This extra target information induces an imitation gap, as the teacher can move directly toward unseen boxes while the student must actively explore to locate them. A box is collected when its distance to the agent is below 1.0m. Episodes terminate when all boxes are collected or when the agent collides with a wall, and are truncated after 200 steps. The reward function is $r_t = 10 \times \mathbf{1}\{\text{all boxes collected}\} + 1 \times \mathbf{1}\{\text{one box collected}\} - 5 \times \mathbf{1}\{\text{wall collision}\} - 0.05$. Success is defined as the fraction of collected boxes, with terminated episodes assigned zero success.

QuarterCheetah is a continuous-control task based on HalfCheetah [30], simulated in MuJoCo [29]. The agent must maximize forward velocity using a 6-dimensional continuous action space, $\mathcal{A} \subset \mathbb{R}^6$. The teacher observes the full 17-dimensional proprioceptive state (joint positions and velocities), while the student observes only the right half of a stack of 4 RGB renders, each resized to 84×84 and normalized to $[-1, 1]$, yielding $\dim(o^T) = 17$ and $\dim(o^S) = 4 \times 3 \times 84 \times 84$. This asymmetry in the observations induces an imitation gap, as the teacher can act directly on full state information while the student must find a working policy while seeing only the right half of the visual input. Episodes follow the default HalfCheetah horizon (1000 steps). The reward function is the standard MuJoCo HalfCheetah reward: $r_t = v_t - 0.1\|a_t\|^2$, where v_t is forward velocity and a_t is the action.

B Zero-shot transfer to new tasks

This appendix provides additional details for the zero-shot transfer experiment described in Section 5.4. The goal of this experiment is to evaluate whether the representation learned by RAIL can be reused to learn new behaviors through teacher-only training and transferred directly to the student agent.

B.1 Modified environments

To evaluate the transfer capabilities of RAIL, we construct modified versions of Corners, TunnelVision, and CollectHealth environments (cf. Appendix A) that support two distinct tasks within the same observation space. The tasks require different behaviors while preserving the same underlying observation manifolds.

CornersPlus. We modify Corners by introducing an additional set of target locations at the midpoints of each wall. At the start of each episode, both a corner target and a side target are sampled uniformly and revealed only to the teacher. In Task 1, the agent must reach the corner target while avoiding side targets; in Task 2, it must reach the side target while avoiding corner targets. To ensure that the required behaviors differ substantially, visiting a side target in Task 1 or a corner target in Task 2 immediately terminates the episode as failure. The modified reward function is $r_t = 1 \times \mathbf{1}\{\text{goal reached}\} - 0.1 \times \mathbf{1}\{\text{termination}\} - 0.01$.

TunnelVisionPlus. We modify TunnelVision by changing the action space while keeping observations unchanged. In Task 1, as in the original TunnelVision, the agent operates with 8 actions (four movements and four rotations). In Task 2, four diagonal movement actions are added, increasing the action space to 12. The reward and termination conditions remain identical to TunnelVision.

CollectHealthPlus. We modify CollectHealth by adding an equal number of secondary objects into the environment (rubber ducks) and changing the rewarded object class. In Task 1, medkits are rewarded and ducks penalized; in Task 2, ducks are rewarded and medkits penalized. Both medkits and ducks are randomly placed with a minimum separation of 3.0m, and the agent is initialized at least 3.0m away from all objects. Unlike the original CollectHealth, we place only two objects of each type to reduce scene density and avoid trivial collection strategies. The modified reward function is $r_t = 10 \times \mathbf{1}\{\text{all target objects collected}\} + 1 \times \mathbf{1}\{\text{target collected}\} - 1 \times \mathbf{1}\{\text{incorrect object collected} \vee \text{wall collision}\} - 0.01$.

B.2 Transfer results

Table 3 shows that the representations learned by RAIL are not tied to a single policy or reward function, but instead capture rich environment details that support new behaviors without additional student-side training. In all three environments, training a new policy on top of a frozen representation learned from Task 1 yields nearly identical student performance to training RAIL from scratch on Task 2. This is notable because, after freezing the encoders, training the new policy is performed entirely using teacher observations. Nevertheless, the resulting policies remain directly executable from student observations, which indicate that the learned latent space preserves the shared environment structure required for learning new behaviors while maintaining teacher–student alignment.

Table 3: Student success rates on the new task. Task 2 denotes direct training on the second task from scratch. Task $1 \to 2$ denotes transfer after first learning the representation on Task 1, freezing both encoders, training a new Task 2 policy using only teacher observations, and deploying zero-shot to the student. Matching Task $1 \to 2$ to Task 2 indicates successful representation reuse. Rates show the mean with upper and lower semideviations across five runs.

	CornersPlus		**TunnelVisionPlus**		**CollectHealthPlus**	
Method	Task 2	Task 1→2	Task 2	Task 1→2	Task 2	Task 1→2
RAIL (student)	$1.0^{+0.00}_{-0.00}$	$1.0^{+0.00}_{-0.00}$	$1.0^{+0.00}_{-0.00}$	$1.0^{+0.00}_{-0.00}$	$0.84^{+0.07}_{-0.07}$	$0.83^{+0.06}_{-0.06}$
Student RL	$1.0^{+0.00}_{-0.00}$	–	$1.0^{+0.00}_{-0.00}$	–	$0.81^{+0.11}_{-0.11}$	–

C Implementation details

C.1 InfoNCE implementation

We use a symmetric InfoNCE objective following the bidirectional formulation of CLIP [19]. Given minibatch embeddings $\{z_T^i, z_S^i\}_{i=1}^N$, we first ℓ_2-normalize both embedding sets and compute similarity logits

$$M_{ij} = s\,(z_T^i)^\top z_S^j, \tag{15}$$

where $s = \exp(\alpha) = 1/\tau$ is a learnable logit scale with unconstrained parameter α. We initialize $s = 1/0.05$, optimize α jointly with all parameters, and clamp $s \leq 50$ for numerical stability.

The contrastive objective averages teacher-to-student and student-to-teacher cross-entropies:

$$\mathcal{L}_{\text{InfoNCE}} = \frac{1}{2}\Big[\text{CE}(M, y) + \text{CE}(M^\top, y)\Big], \tag{16}$$

where CE is the cross entropy loss and $y = (1, \ldots, N)$ specifies that sample i is matched to its paired sample i.

Multi-positive variant. In environments with small or discrete observation spaces, multiple samples in a minibatch may correspond to identical teacher or student observations. Treating these duplicates as negatives creates false-negative pairs. We therefore replace the one-hot target with a multi-positive mask $P \in \{0, 1\}^{N \times N}$:

$$P_{ij} = \mathbf{1}\Big[o_T^i = o_T^j \;\vee\; o_S^i = o_S^j\Big]\,. \tag{17}$$

The teacher-to-student term becomes

$$\mathcal{L}_{T\to S} = -\frac{1}{N}\sum_{i=1}^{N}\log\sum_{j=1}^{N} P_{ij}\frac{\exp(M_{ij})}{\sum_{k=1}^{N}\exp(M_{ik})}, \tag{18}$$

and the reverse-direction term is

$$\mathcal{L}_{S\to T} = -\frac{1}{N}\sum_{i=1}^{N}\log\sum_{j=1}^{N} P_{ji}\frac{\exp(M_{ji})}{\sum_{k=1}^{N}\exp(M_{ki})}. \tag{19}$$

The final objective is

$$\mathcal{L}_{\text{MP-InfoNCE}} = \frac{1}{2}\left(\mathcal{L}_{T\to S} + \mathcal{L}_{S\to T}\right). \tag{20}$$

We use this variant only in Corners and TunnelVision, where repeated observations are common.

C.2 Network architectures

All methods use comparable model capacity. RAIL consists of a teacher and a student encoder, a shared encoder, and a policy network with actor–critic heads.

Vector observations. For state-based agents, each observation encoder is a three-layer MLP with hidden width 512 and LeakyReLU activations: $d_{\text{obs}} \to 512 \to 512 \to 512$.

Pixel observations. For pixel observations, we use a four-layer CNN with 3×3 kernels, 32 channels per layer, and strides $(2, 2, 1, 1)$, followed by a linear layer producing a 512-dimensional feature vector.

Shared encoder. Teacher and student features are concatenated with a learned agent-identity embedding of dimension 4, then mapped by a shared MLP: $516 \to 512 \to 512 \to d_z$, followed by ℓ_2-normalization for contrastive learning.

Policy network. We use the standard PPO policy architecture: two shared hidden layers of size 64 with ReLU activations, followed by separate linear actor and critic output heads.

C.3 Training hyperparameters

We use PPO for all RL-based methods and Adam for both policy and encoder optimization. Table 4 lists the environment-specific training budgets and hyperparameters. Unless stated otherwise, all remaining hyperparameters use the default values from PyTorch [16] and Stable-Baselines3 [20].

For QuarterCheetah, we found that direct Student RL and the privileged teacher training used for BC became unstable with the standard PPO learning rate. We therefore reduced the PPO learning rate to 2×10^{-5} for these settings to prevent training collapse.

All experiments were conducted on a machine with an Intel Core i7-13700KF CPU (16 cores, 24 threads, up to 5.4 GHz), an NVIDIA RTX 4070 GPU with 12 GB VRAM, and 32 GB RAM. Depending on the environment and method, individual runs required approximately 1–7 hours. In QuarterCheetah, RAIL required approximately four hours, compared with seven hours for standard PPO from pixels under the same environment-interaction budget.

The BC datasets are collected from rollouts of the trained state-based teacher and contain 4,096 transitions for Corners and TunnelVision, 20,000 for CollectHealth, and 80,000 for QuarterCheetah. DAgger uses replay buffers of the same respective sizes, populated from student rollouts.

Table 4: Training hyperparameters for each environment.

	Corners	TunnelVision	CollectHealth	QuarterCheetah
training steps	1.0M	1.5M	0.4M	2.5M
parallel environments	12	12	12	4
PPO				
optimizer	Adam	Adam	Adam	Adam
learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}
batch size	256	512	256	64
rollout steps (n_{steps})	256	512	256	512
epochs	10	10	10	20
entropy coefficient	0.002	0.001	0.000	0.0004
clip range	0.1	0.1	0.1	0.1
discount factor (γ)	0.99	0.99	0.99	0.98
RAIL				
encoder optimizer	Adam	Adam	Adam	Adam
encoder learning rate	10^{-3}	10^{-3}	10^{-3}	10^{-3}
embedding dimension (d_z)	64	128	256	256
contrastive batch size	512	512	256	512
contrastive iterations (K)	1	2	12	8
λ_{con}	1.0	1.0	1.0	1.0
λ_{align}	1.0	1.0	1.0	1.0
λ_{stab}	1.0	1.0	1.0	1.0
initial InfoNCE scale	20	20	20	20
maximum InfoNCE scale	50	50	50	50

D Additional experiments

D.1 Privileged-information probes

We use Corners to test whether teacher embeddings retain shared information while suppressing information private to the teacher. The target corner is sampled uniformly from four possibilities and observed only by the teacher, whereas the agent's coordinates are available to both agents. We train RAIL from scratch with ten random seeds and freeze the learned encoders. For each model, we collect probe data from 100 independently initialized episodes with independently sampled target corners.

We train separate two-layer MLP probes on the frozen teacher embeddings to predict the target corner and the agent's coordinates. We evaluate target-corner prediction only at the initial state, where the agent always starts at the grid center. This prevents the probe from exploiting correlations between the agent's position and target corner that arise during a rollout. The coordinate probe uses embeddings from the full rollout buffer to assess whether the representation preserves shared spatial information.

Across ten seeds, target-corner prediction achieves $25.4 \pm 1.4\%$ accuracy, close to the 25% chance level, while coordinate prediction achieves $100.0 \pm 0.0\%$. These results provide evidence that RAIL preserves shared information while suppressing information private to the teacher.

D.2 Contrastive batch size

The number of negatives in our contrastive objective (Eq. (11)) depends on the batch size used for representation learning. We experiment with batch sizes of 128, 256, 512, and 1024, keeping the remaining training setup fixed and using five random seeds per setting. Table 5 reports student performance.

RAIL achieves high student performance throughout this range. In Corners, batch sizes of 512 and 1024 yield perfect success in all five runs, whereas smaller batches have lower mean success and greater variability. TunnelVision and CollectHealth approach saturation at moderate batch sizes, and QuarterCheetah returns remain similar across settings. Thus, larger batches can improve reliability, while RAIL maintains strong performance even with smaller batches.

Table 5: Contrastive batch-size ablation. Student success rates show the mean with upper and lower semideviations over five seeds; QuarterCheetah episodic returns show mean $\pm$ standard deviation.

Batch size	Corners	TunnelVision	CollectHealth	QuarterCheetah
128	$0.91^{+0.08}_{-0.16}$	$0.99^{+0.01}_{-0.01}$	$0.94^{+0.03}_{-0.04}$	2491.33 ± 154.06
256	$0.89^{+0.08}_{-0.10}$	$1.00^{+0.00}_{-0.00}$	$0.98^{+0.00}_{-0.01}$	2523.33 ± 84.63
512	$1.00^{+0.00}_{-0.00}$	$1.00^{+0.00}_{-0.00}$	$0.99^{+0.00}_{-0.01}$	2510.24 ± 72.05
1024	$1.00^{+0.00}_{-0.00}$	$1.00^{+0.00}_{-0.00}$	$0.99^{+0.01}_{-0.01}$	2543.33 ± 45.76

D.3 Baseline experiments under matched observations

To test whether the failure of the baselines arises from information mismatch, we repeat BC, DAgger, COSIL, and SITT with the same training protocols but matched information content. Students receive the teacher's observations in Corners, TunnelVision, and CollectHealth. In QuarterCheetah, students receive stacked, unmasked RGB images while teachers retain state observations.

All four methods match or approach teacher performance across all environments (Table 6). This contrast with Table 2 supports that the information mismatch is a principal contributor to the failure of baselines in our environments.

Table 6: Baseline performance under matched observations. Asterisks denote environments with modified student observations. Teacher (T) and student (S) success rates show the mean with upper and lower semideviations over five seeds; QuarterCheetah returns show mean $\pm$ standard deviation. The imitation gap is $\Delta = T - S$. Higher S and lower Δ are better.

	Corners*			**TunnelVision***			**CollectHealth***			**QuarterCheetah***		
Method	T	$S\uparrow$	$\Delta\downarrow$	T	$S\uparrow$	$\Delta\downarrow$	T	$S\uparrow$	$\Delta\downarrow$	T	$S\uparrow$	$\Delta\downarrow$
BC	$1.00^{+0.00}_{-0.00}$	$1.00^{+0.00}_{-0.00}$	0.00	$1.00^{+0.00}_{-0.00}$	$1.00^{+0.00}_{-0.00}$	0.00	$1.00^{+0.00}_{-0.00}$	$1.00^{+0.00}_{-0.00}$	0.00	2186.83 ± 82.47	2113.43 ± 57.39	73.40
DAgger	$1.00^{+0.00}_{-0.00}$	$1.00^{+0.00}_{-0.00}$	0.00	$1.00^{+0.00}_{-0.00}$	$1.00^{+0.00}_{-0.00}$	0.00	$1.00^{+0.00}_{-0.00}$	$1.00^{+0.00}_{-0.00}$	0.00	2186.83 ± 82.47	2099.67 ± 62.27	87.16
COSIL	$1.00^{+0.00}_{-0.00}$	$1.00^{+0.00}_{-0.00}$	0.00	$1.00^{+0.00}_{-0.00}$	$1.00^{+0.00}_{-0.00}$	0.00	$1.00^{+0.00}_{-0.00}$	$1.00^{+0.00}_{-0.00}$	0.00	2186.83 ± 82.47	2090.52 ± 68.50	96.31
SITT	$1.00^{+0.00}_{-0.00}$	$1.00^{+0.00}_{-0.00}$	0.00	$1.00^{+0.00}_{-0.00}$	$1.00^{+0.00}_{-0.00}$	0.00	$1.00^{+0.00}_{-0.00}$	$1.00^{+0.00}_{-0.00}$	0.00	1865.42 ± 86.72	1720.43 ± 74.45	144.99

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately describe the paper's main contributions, including the proposed RAIL framework, the representation-learning objective, and the empirical evaluation on asymmetric-observation reinforcement learning tasks. The claims are supported by the experimental results and are limited to the evaluated settings.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have an explicit subsection in Section 6 addressing this point.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present formal theoretical results such as theorems, propositions, or proofs. The discussion of the imitation gap and conditional independence assumptions is intended to provide intuition and motivation for the proposed method rather than formal guarantees.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides detailed descriptions of the environments, observation spaces, network architectures, training procedures, PPO hyperparameters, contrastive learning setup, and evaluation protocols in both the main text and appendix. The paper also reports training budgets, batch sizes, random seed evaluation settings, and implementation details necessary to reproduce the experimental results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The implementation and instructions are available at https://github.com/meraccos/RAIL.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the environments, observation spaces, model architectures, optimization procedures, PPO hyperparameters, contrastive learning setup, training budgets, and evaluation protocols necessary to understand the reported results. Additional implementation details are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The paper reports averages over multiple random seeds and evaluation environments, and the learning curves include shaded error regions across runs. The evaluation protocol and number of seeds are specified in the paper and appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: These are specified in Appendix C.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work complies with the NeurIPS Code of Ethics. The research is methodological and evaluated in simulated environments, with no collection or use of sensitive personal data. The proposed methods do not pose immediate risks of misuse beyond those typical for reinforcement learning methods.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The paper does not explicitly discuss broader societal impacts. The work is primarily methodological and focuses on improving teacher–student transfer in reinforcement learning under asymmetric observations. Potential risks and benefits are similar to those already associated with existing reinforcement learning and autonomous decision-making systems, rather than introducing new societal concerns specific to the proposed method.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not pose such risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use publicly available existing implementations of baseline methods and training environments, which are properly cited. We rely on standard public implementations and do not redistribute any third-party assets.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The RAIL implementation is available at `https://github.com/meraccos/RAIL`.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: the paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: the research does not involve LLMs as an important, original, or non-standard component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.