EMBEDDING INFORMATION IN POLICY RANDOMNESS: DISTRIBUTION-PRESERVING IMPLICIT COMMUNICATION IN MDPS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper investigates a new role of randomness in stochastic policies—as a medium for embedding and communicating information. Specifically, we consider an agent interacting with an environment according to a prescribed stochastic policy. By carefully modulating the random action realizations of the policy, information can be embedded into the agent's behavior without altering the prescribed action distribution. We cast this problem as a channel coding problem, in which the controlled system implicitly serves as a communication channel. We then propose a coding scheme, termed posterior-matching randomness modulation (PRAM), that selects actions according to the message to be transmitted. With the aid of common randomness independent of the control system, PRAM preserves the prescribed action distribution for every state and every message. We further prove that PRAM is asymptotically reliable: the message can be decoded with arbitrarily small error probability as the observed state trajectory becomes sufficiently long. Experiments demonstrate the effectiveness of the proposed framework in applications including multi-bit policy watermarking and trajectory provenance.

1 INTRODUCTION

Stochastic policies are widely used in control systems and reinforcement learning (RL), where policy randomness facilitates exploration (Ahmed et al., 2019), enhance robustness (Eysenbach & Levine, 2021), and supports multimodal decision making (Zhao et al., 2019). Under a stochastic policy, actions are sampled from a state-dependent distribution rather than selected deterministically. This paper reveals a new role of policy randomness: it can be exploited to embed information into an agent's behavior without degrading its control performance. Specifically, we consider a Markov decision process (MDP) in which an agent interacts with the environment according to a prescribed stochastic policy (Puterman, 2014). Our goal is to enable the agent to embed and communicate a message through its action realizations while preserving the prescribed action distribution at every state. In other words, the embedded information should be recoverable from the resulting state trajectory without changing the marginal behavior of the reference policy. We refer to this technique as *distribution-preserving implicit communication (DPIC)*.

One important application of this technique is multi-bit policy watermarking. RL has been increasingly deployed in practical decision-making systems, ranging from robotics and autonomous systems to energy management, inventory control, and cyber-defence (Liu et al., 2025; Yu et al., 2021; Nguyen & Reddi, 2021). In many such applications, the resulting policies are trained using valuable data and substantial computational and engineering resources, making them important intellectual property (IP) of the development team. Protecting the IP of these RL policies therefore requires mechanisms that can verify policy ownership and detect unauthorized use or replication. Motivated by this need, recent studies (Amir et al., 2026; Chen et al., 2021; Behzadan & Hsu, 2019; Huang et al., 2025) have investigated digital watermarking methods for policy IP protection. However, existing methods primarily support zero-bit watermarking, where the detector only determines whether a watermark is present. In contrast, the DPIC framework naturally enables multi-bit watermarking, allowing policy owners to embed richer information for identification and attribution. The same information-embedding mechanism can also support closely related tasks such as trajectory

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

provenance (Gao et al., 2026) and more general behavioral metadata embedding (An et al., 2026) for source attribution, authentication, and traceability.

As suggested by its name, DPIC casts information embedding as a communication problem, and more specifically, a channel coding problem. Inspired by implicit communication in control systems (Chen & Gündüz, 2025; Wu et al., 2025), we view the MDP environment as a state-dependent communication channel, where the agent's action serves as the channel input and the resulting state serves as the channel output. Under this interpretation, embedding a B-bit message into the agent's behavior is equivalent to transmitting the message through this channel.

A key objective of DPIC is to enable such communication without degrading the original control performance. This motivates a distribution-preserving requirement on the agent's actions. Unlike conventional communication, where the input distribution is typically constrained only after averaging over the message distribution (Cover et al., 1991), applications such as policy watermarking may repeatedly embed the same fixed message. Message-averaged preservation is therefore insufficient to guarantee unchanged control performance. We instead require the action distribution to match the prescribed stochastic policy for every state and every message. This requirement poses a fundamental challenge. From an information-theoretic perspective, preserving the same action distribution for every message implies that, conditioned on the current state, the action alone is independent of the message and therefore cannot directly convey information about it. To overcome this limitation, we introduce common randomness shared between the agent (encoder) and the observer (decoder), and select each action based on the message, the observed state history, and the shared random variable. Communication is then enabled through the message-dependent coupling between the action and the common randomness, while the marginal action distribution remains unchanged for every message.

Building on this idea, we develop the *posterior-matching randomness modulation (PRAM)* scheme. The term *randomness modulation* refers to encoding information by controlling the realization, rather than the distribution, of policy randomness. We show that PRAM preserves the prescribed action distribution for every state and every message, while still enabling reliable information transmission through the MDP dynamics. Under mild conditions, the posterior probability of the transmitted message converges to one as the trajectory length increases, ensuring asymptotically reliable message recovery. We further demonstrate that PRAM remains effective under model mismatch, where the encoder and decoder rely on an estimated transition kernel rather than the true system dynamics, making the method particularly relevant to learning-based systems.

Our main contributions are summarized as follows. **(i)** We introduce the idea of embedding information in policy randomness and formulate it as DPIC. **(ii)** We develop a posterior-matching randomness modulation scheme based on common randomness and prove its per-message distribution-preserving property. **(iii)** We establish the asymptotic reliability of PRAM. **(iv)** We demonstrate the effectiveness of PRAM in multi-bit policy watermarking and trajectory provenance, including under model mismatch.

2 Related Work

Policy watermarking. Existing policy-watermarking methods mainly focus on ownership verification through behavioral signatures. Early work embeds watermarks through designated state-action patterns (Behzadan & Hsu, 2019), while temporal watermarking exploits sequential behaviors for more robust verification (Chen et al., 2021). More recent approaches extend behavioral watermarking to general agents (Huang et al., 2025) and remotely observable robotic policies (Amir et al., 2026). These methods are primarily designed for zero-bit watermark detection, where the detector can only distinguish between the presence and absence of a watermark. In contrast, multi-bit policy watermarking remains largely unexplored. A closely related work on multi-bit behavioral watermarking for LLM-based agents is TRACE (Gao et al., 2026), which also exploits controlled action sampling to embed information. However, its detector relies on direct observation of the executed actions, which is generally unavailable in many MDP and control settings. We therefore consider a more realistic setting in which the observer only observes the resulting state trajectory, and the embedded information must be conveyed through the stochastic system dynamics.

Sampling-based watermarking for generative models. Compared with policy watermarking, multi-bit watermarking for generative models, particularly LLMs, has received substantially

more attention. Existing methods embed information by modifying or controlling the token-sampling process while attempting to preserve text quality and statistical properties of the original model (Kirchenbauer et al., 2023; Yoo et al., 2024; Wang et al., 2024; Park et al., 2026; Zhu et al., 2026). However, by the nature of autoregressive LLM generation, these methods assume that the generated tokens—the actions under a sequential decision interpretation—are directly observable to the detector. As in (Gao et al., 2026), this leads to a fundamentally different decoding setting from ours and prevents these methods from being applied to general policy watermarking problems.

Implicit communication and feedback coding. The idea of DPIC is closely related to communicating via MDPs (Sokota et al., 2022; Karabag et al., 2019) and implicit communication in control systems (Chen & Gündüz, 2025; 2026; Rips & Sabag, 2025). These works view the controlled system as a communication channel and mainly characterize the tradeoff between communication rate and control performance across different policies. The most relevant work is Act2Comm (Wu et al., 2025), which studies communication through actions in finite-state MDPs, a setting similar to ours. Essentially, Act2Comm addresses the problem of channel coding over a finite-state channel with feedback and, like conventional channel coding, constrains the input distribution only after averaging over the message distribution. In contrast, DPIC imposes a stronger requirement: the prescribed action distribution must be preserved for every state and every message. This distinction is particularly important for applications such as policy watermarking, where the same identifier may be embedded repeatedly, and makes the DPIC problem fundamentally more restrictive.

Nevertheless, our method is inspired by feedback channel coding (Gallager & Nakiboğlu, 2010; Tatikonda & Mitter, 2008; Permuter et al., 2009), particularly posterior matching (Shayevitz & Feder, 2011) and its variant (Li & El Gamal, 2015). With feedback, the encoder observes the same channel outputs as the decoder and can therefore track the decoder's posterior belief over the message. Posterior matching exploits this shared posterior to adaptively determine what information should be conveyed next. Building on this principle, we introduce common randomness shared by the encoder and decoder and develop a posterior-matching-based scheme tailored to DPIC. The common randomness enables message-dependent action selection while preserving the prescribed action distribution for every message, which is the key distinction from conventional schemes.

Notations. We use s^t to denote the sequence $(s_1, s_2, \ldots, s_t)$. When it is necessary to distinguish a random variable from its realization, we use boldface notation, e.g., $\mathbf{m}$, for the random variable and the corresponding regular font, e.g., m, for its realization.

3 Problem Statement

3.1 System Model

Consider a system in which an *agent* interacts with its environment by taking actions. We model the system as an MDP defined by a tuple $(\mathcal{S}, \mathcal{A}, r, T)$, where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is a finite action space, $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition kernel, and $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function. For simplicity, we index the actions by letting $\mathcal{A} = \{1, 2, \cdots, |\mathcal{A}|\}$. At each time t, the agent observes the system state s_t and selects an action a_t. The environment then transitions to a new state s_{t+1} according to the distribution $T(\cdot|s_t, a_t)$. We further assume the presence of a *passive observer* that can observe the state s_t but cannot take actions or otherwise influence the environment. The system model is illustrated in Fig. 1.

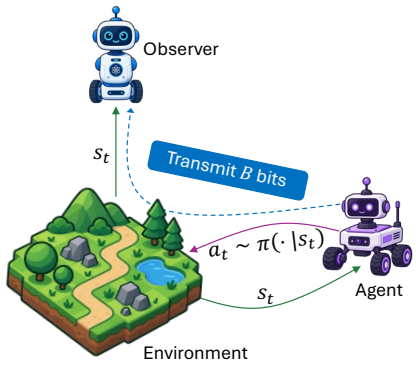


Figure 1: System model.

A stationary control policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ is a mapping that specifies an action distribution for each state, where $\Delta(\mathcal{A})$ denote the probability simplex over $\mathcal{A}$. We use $\pi(a|s)$ to denote the probability that policy π selects action $a \in \mathcal{A}$ in state

$s \in \mathcal{S}$. Unlike standard MDP or RL problems, our objective is not to identify an optimal policy that maximizes the cumulative or average reward. Instead, we assume that the agent is already equipped with a suitable stationary policy π, referred to as the *reference policy*. Our central question is: *Can the agent encode a message in its actions while following π, such that the observer can accurately decode the message from the resulting sequence of observed states?*

In particular, let $\mathbf{m}$ denote a message drawn from a finite message set $\mathcal{M} = \{0, 1, \cdots, 2^B - 1\}$ according to a distribution $P_{\mathbf{m}}$, where B denotes the message length in bits. Conditioned on $\mathbf{m}$, the agent selects an action sequence $a_{1:n}$, with each action a_t determined using the message $\mathbf{m}$ and the state history up to the current time step t. After observing the resulting state sequence $s_{1:n+1}$, the observer produces an estimate $\hat{\mathbf{m}}$ of the message. The estimation error probability, also referred to as the decoding error probability, is then defined as $P_e^{(n)} \triangleq P(\hat{\mathbf{m}} \neq \mathbf{m})$.

Our objective is twofold: **(i)** to minimize $P_e^{(n)}$ for a fixed horizon n, and **(ii)** to ensure, for every realization of $\mathbf{m}$, that the agent's action at each time step remains distributed according to the reference policy, i.e., $\mathbf{a}_t \sim \pi(\cdot|\mathbf{s}_t), \forall t$. This model has applications in a range of practical settings, including multi-bit policy watermarking and trajectory provenance, as will be demonstrated in greater detail in the experimental section.

3.2 Problem Formulation

In essence, our task is to enable the agent to transmit a message to the observer without relying on an external communication channel, instead using the MDP environment itself as the channel. This problem can be studied through the lens of implicit communication (Chen & Gündüz, 2025). Specifically, the MDP environment can be interpreted as a finite-state channel (FSC) between the agent (encoder) and the observer (decoder): at each time step t, the action a_t serves as the channel input, and the current state s_t serves as the channel state. Upon execution of a_t, the environment transitions to a new state s_{t+1}, which constitutes the corresponding channel output. Under this interpretation, the reference policy π specifies the desired input distribution of the FSC: conditional on the current channel state s_t, the channel input should be distributed according to $\pi(\cdot|s_t)$.

Within the framework of implicit communication, the problem reduces to designing a channel code for the FSC induced by the MDP. In channel coding, such a code consists of an encoder and a decoder. The encoder is a sequence of mappings $\mathcal{E} = \{\mathcal{E}_t : 1 \leq t \leq n\}$, where $\mathcal{E}_t : \mathcal{M} \times \mathcal{S}^t \times \mathcal{A}^{t-1} \to \mathcal{A}$ determines the action a_t from the message, the state history, and the past actions. After observing the initial channel state s_1 and the subsequent n channel outputs $s_{2:n+1}$, the observer applies the decoder mapping $\mathcal{D} : \mathcal{S}^{n+1} \to \mathcal{M}$ to estimate the transmitted message. Here, n denotes the *blocklength*, corresponding to the number of time steps used to transmit the message.

Conventional channel coding requires the channel input to follow a prescribed distribution, such as a capacity-achieving distribution, only after averaging over all messages. Our setting, however, imposes a stronger constraint: for every realization m of $\mathbf{m}$, the channel input at time t must follow the prescribed distribution $\pi(\cdot|s_t)$. At first glance, this requirement appears to preclude communication. Indeed, if $P(a_t|s_t, m) = \pi(a_t|s_t)$ for every m, then $\mathbf{a}_t$ and $\mathbf{m}$ are conditionally independent given s_t, and hence the conditional mutual information $I(\mathbf{a}_t; \mathbf{m}|s_t) = 0$. As a result, input $\mathbf{a}_t$ conveys no additional information about the message beyond that already contained in the current state.

To address this challenge, we introduce common randomness shared between the agent and the observer. Specifically, let $\{\mathbf{u}_t \in \mathcal{U} : t \geq 1\}$ be a sequence of independent and identically distributed (i.i.d.) random variables, with each $\mathbf{u}_t$ drawn from a distribution $P_{\mathbf{u}}$. At each time t, the realization of $\mathbf{u}_t$ is revealed to both the agent and the observer. We assume that $\mathbf{u}_t$ is independent of the message $\mathbf{m}$, the state history $\mathbf{s}^t$, and the action history $\mathbf{a}^{t-1}$. In practice, such common randomness can be generated using a shared random seed derived from a secret key.

With the introduction of common randomness, we redefine the encoder as

$$\mathcal{E}_t : \mathcal{M} \times \mathcal{S}^t \times \mathcal{A}^{t-1} \times \mathcal{U}^t \to \mathcal{A},\ t \geq 1.$$

Thus, at time t, the agent uses the shared random variables $\mathbf{u}^t$, together with the message and the available state and action histories, to determine the action $\mathbf{a}_t$. Correspondingly, the decoder is defined as $\mathcal{D} : \mathcal{S}^{n+1} \times \mathcal{U}^n \to \mathcal{M}$. The common randomness makes communication possible because the encoder can satisfy $I(\mathbf{a}_t; \mathbf{m}|s_t) = 0$ while maintaining $I(\mathbf{a}_t; \mathbf{m}|s_t, u_t) > 0$. In other

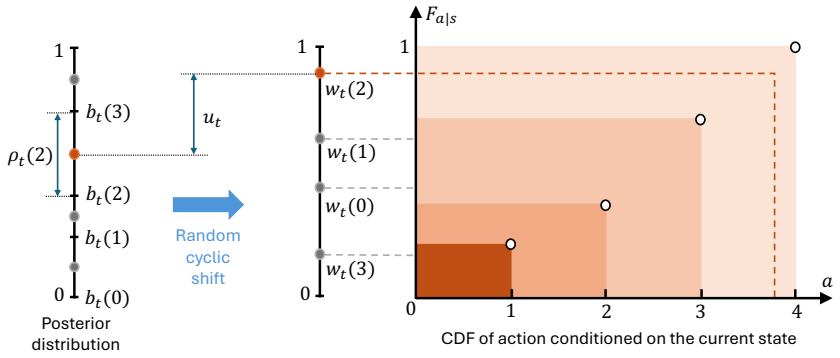


Figure 2: An illustration of the coding algorithm. Suppose $|\mathcal{M}| = 4$ and the message is $m^* = 2$, then $a_t(m^*) = 4$ in this example.

words, although the action reveals no information about the message when conditioned only on the current state, it can depend strongly on the message once the shared randomness is known. Our objective is to design an encoder–decoder pair $(\mathcal{E}, \mathcal{D})$ that enables efficient message transmission while ensuring that the agent adheres to the reference policy π.

4 Distribution-preserving Implicit Communication

In this section, we introduce a coding scheme that preserves the conditional action distribution $\mathbf{a}_t \sim \pi(\cdot|\mathbf{s}_t)$ for every message m. We also provide a theoretical guarantee that, under this scheme, the observer's decoding error probability vanishes as the blocklength n tends to infinity.

4.1 Coding Scheme

Our method is largely inspired by posterior matching (Shayevitz & Feder, 2011) and its variant (Li & El Gamal, 2015), both of which are coding schemes for feedback communication. However, these schemes preserve the desired input distribution only after averaging over all messages, rather than for each message individually. To overcome this limitation, we introduce the common randomness $\mathbf{u}_t$ and develop a novel coding scheme that leverages it to preserve the prescribed input distribution $\pi(\cdot|\mathbf{s}_t)$ for every message.

We first adapt the posterior matching principles introduced by Shayevitz & Feder (2011) to incorporate common randomness. At each time t, the observer has access to the state sequence $\mathbf{s}^t$ and the common random sequence $\mathbf{u}^{t-1}$, which together provide information about the message $\mathbf{m}$. Let $\rho_t(\cdot)$ denote the observer's posterior distribution over $\mathbf{m}$, defined as

$$\rho_t(m) \triangleq P(\mathbf{m} = m|\mathbf{s}^t, \mathbf{u}^{t-1}),\ \forall m \in \mathcal{M}.$$

Specifically, $\rho_1(m) = P_{\mathbf{m}}(m)$ as $\mathbf{m}$ and $\mathbf{s}_1$ are independent. As long as ρ_t is not a point mass, the observer retains some uncertainty about $\mathbf{m}$ and therefore requires additional information to identify it. The posterior matching principle addresses the following question: *given* $\mathbf{s}^t$ *and* $\mathbf{u}^{t-1}$*, what additional information is needed to determine* $\mathbf{m}$*?* A natural candidate is a random variable $\mathbf{w}_t$ satisfying the following conditions: (i) $\mathbf{w}_t$ is independent of $(\mathbf{s}^t, \mathbf{u}^{t-1})$; and (ii) the message $\mathbf{m}$ can be uniquely determined by $(\mathbf{w}_t, \mathbf{s}^t, \mathbf{u}^t)$. They are referred to as two *design principles*.

Intuitively, the first principle ensures that $\mathbf{w}_t$ does not convey any "old information" already available to the observer, while the second guarantees that $(\mathbf{w}_t, \mathbf{u}_t)$ carries sufficient "new information" to determine the message. Given an appropriate construction of $\mathbf{w}_t$, the action $\mathbf{a}_t$ is then selected as a function of $\mathbf{w}_t$ while ensuring that its conditional distribution matches the reference policy $\pi(\cdot \mid \mathbf{s}_t)$.

Following this idea, we are now ready to introduce our coding scheme. First, define

$$b_t(m) \triangleq P(\mathbf{m} < m|\mathbf{s}^t, \mathbf{u}^{t-1}) = \sum_{j<m} \rho_t(j).$$

Note that $(\mathbf{s}^t, \mathbf{u}^{t-1})$ is also available to the agent; therefore, the agent can also compute ρ_t and b_t at each time. We then construct the information point at time t as follows:

$$\mathbf{w}_t(\mathbf{m}) = [b_t(\mathbf{m}) + l_t(\mathbf{m})\rho_t(\mathbf{m}) + \mathbf{u}_t] \mod 1,$$

where $l_t : \mathcal{M} \to (0, 1)$ is a deterministic function and $\mathbf{u}_t \sim \text{Unif}[0, 1]$ independent of all the other variables. For each $m \in \mathcal{M}$, the quantity $b_t(m)+l_t(m)\rho_t(m)$ lies within the interval $(b_t(m), b_t(m+1))$, and $\mathbf{w}_t(m)$ is obtained by applying a random cyclic shift to this point, as illustrated in Figure 2. We write $\mathbf{w}_t$ in place of $\mathbf{w}_t(\mathbf{m})$ when referring to the random variable induced by $\mathbf{m}$, rather than to its value for a particular realization of $\mathbf{m}$. The choice of l_t determines the pairwise distances between information points and thus affects the observer's decoding error probability. In this paper, we set $l_t(m) = 0.5$ for all $m \in \mathcal{M}$, as this choice simplifies the reliability analysis.

It is straightforward to verify that $\mathbf{w}_t$ satisfies the two design principles stated above. First, for every $m \in \mathcal{M}$ and any realization (s^t, u^{t-1}), we have $\mathbf{w}_t(m) \sim \text{Unif}[0, 1]$, which implies that $\mathbf{w}_t$ is independent of $(\mathbf{s}^t, \mathbf{u}^{t-1})$. Second, the message $\mathbf{m}$ can be uniquely recovered from $\mathbf{w}_t$, $\mathbf{u}_t$, ρ_t, and b_t. Because ρ_t and b_t are determined by $(\mathbf{s}^t, \mathbf{u}^{t-1})$, the second design principle also holds.

The next step is to select an action based on $\mathbf{w}_t$ while preserving the desired distribution. To this end, let $F_{\mathbf{a}_t|\mathbf{s}_t}(\cdot)$ denote the cumulative distribution function (CDF) of $\pi(\cdot|\mathbf{s}_t)$. Its generalized inverse is defined as

$$F^{-1}_{\mathbf{a}_t|\mathbf{s}_t}(w) := \inf\{a \in \mathcal{A} : F_{\mathbf{a}_t|\mathbf{s}_t}(a) \geq w\}, \forall w \in [0, 1].$$

The action at time t is then determined by $\mathbf{a}_t = F^{-1}_{\mathbf{a}_t|\mathbf{s}_t}(\mathbf{w}_t)$. Upon execution of $\mathbf{a}_t$, the MDP transitions to a new state $\mathbf{s}_{t+1}$, which is observed by both the agent and the observer. Given any realization s^{t+1}, the posterior distribution is then updated according to Bayes' rule:

$$\rho_{t+1}(m) = \frac{\rho_t(m)T(s_{t+1}|s_t, a_t(m))}{\sum_j \rho_t(j)T(s_{t+1}|s_t, a_t(j))}, \ \forall m \in \mathcal{M}, \tag{1}$$

where $a_t(m) = F^{-1}_{\mathbf{a}_t|\mathbf{s}_t}(\mathbf{w}_t(m))$. At the end of the transmission horizon, i.e., time $n+1$, the observer can decode the message using the posterior distribution ρ_{n+1}. A natural choice is the maximum a posteriori (MAP) estimator:

$$\hat{\mathbf{m}} = \arg\max_{m \in \mathcal{M}} \rho_{n+1}(m).$$

We refer to this method as *posterior-matching randomness modulation (PRAM)*.

It remains to show that PRAM preserves the prescribed input distribution for every message m; this is formally established in the following result.

Theorem 1 (Distribution Preservation)**.** *Under the PRAM scheme, the following holds for every realization $m \in \mathcal{M}$ of the message $\mathbf{m}$:*

$$P(a_t|s^t, u^{t-1}, m) = \pi(a_t|s_t), \ \forall a_t, s^t.$$

Proof. The proof is provided in Appendix A.1. □

Note that PRAM selects each action $\mathbf{a}_t$ based $\mathbf{s}^t, \mathbf{u}^t$, and $\mathbf{m}$. Hence Theorem 1 means that, when averaged over the common random variable $\mathbf{u}_t$, PRAM preserves the action distribution prescribed by the reference policy. Consequently, the communication capability is achieved without degrading the expected control reward. This distribution-preserving property is a key advantage of PRAM.

4.2 Reliability Analysis

According to the design principles of PRAM, $\mathbf{w}_t$ carries information about the message $\mathbf{m}$, and so does the action $\mathbf{a}_t$, since it is generated as a function of $\mathbf{w}_t$. However, this information is conveyed to the observer only through the stochastic state transitions of the system and may therefore be corrupted by transition noise. It is thus not immediately clear whether PRAM enables reliable message recovery. In this subsection, we establish a theoretical guarantee of asymptotic reliability: the observer's decoding error probability can be made arbitrarily small by increasing the blocklength.

We first introduce the following assumptions, which are essential for our analysis.

Table 1: Total variation (TV) distance between the empirical action distribution induced by PRAM and the corresponding reference policy for different messages.

Environment	Reference Policy	Message							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
ReFrozenLake	Policy 1	0.012	0.012	0.010	0.011	0.011	0.010	0.010	0.009
	Policy 2	0.015	0.011	0.010	0.010	0.011	0.011	0.012	0.011
ConTaxi	Policy 1	0.007	0.006	0.006	0.007	0.006	0.007	0.007	0.006
	Policy 2	0.008	0.008	0.008	0.009	0.008	0.008	0.008	0.008
SBM	Policy 1	0.023	0.023	0.023	0.023	0.023	0.023	0.023	0.023
	Policy 2	0.029	0.028	0.028	0.029	0.029	0.030	0.028	0.029

Assumption 1. *The Markov chain induced by the reference policy π is irreducible.*

Assumption 2. *Under the reference policy π, there exists a state s^* such that $\pi(\cdot|s^*)$ is nondegenerate and $D_{KL}(T(\cdot|s^*, a)||T(\cdot|s^*, a')) > 0$ for any pair of distinct actions a and a' in the support of $\pi(\cdot|s^*)$.*

Irreducibility is a standard assumption in the analysis of finite-state MDPs and naturally yields the property required in Assumption 1. Assumption 2 ensures the existence of at least one recurrent state at which the reference policy is genuinely stochastic and the transition distributions induced by different actions are distinguishable. Now we are ready to present the reliability guarantee.

Theorem 2 (Asymptotic Reliability)**.** *Let $m^* \in \mathcal{M}$ be the message transmitted by the agent. Then, under Assumptions 1 and 2,*

$$\rho_n(m^*) \xrightarrow{\text{a.s.}} 1 \quad \text{as } n \to \infty.$$

As a result, the observer's decoding error probability $P_e^{(n)} \to 0$ as $n \to \infty$.

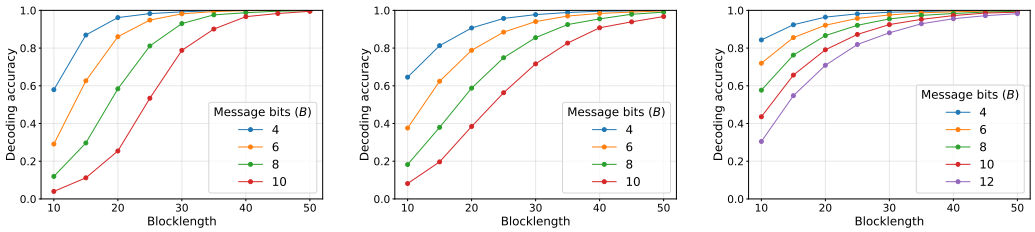
Proof. The proof is provided in Appendix A.2. □

The significance of Theorem 2 lies in the per-message distribution-preserving constraint of PRAM. As discussed earlier, requiring the action distribution to coincide with the reference policy for every message implies $I(\mathbf{a}_t; \mathbf{m}|\mathbf{s}_t) = 0$, meaning that, conditioned on the current state, the action alone reveals no additional information about the message. The introduction of common randomness $\mathbf{u}_t$, however, creates a message-dependent coupling between the action selection and the shared randomization, making communication possible without altering the prescribed action distribution. Theorem 2 confirms that this mechanism is sufficient for reliable communication: despite the per-message distribution-preserving constraint, the observer can recover the transmitted message with vanishing error probability as the blocklength grows.

5 Experiments

This section presents experimental results demonstrating the effectiveness of the proposed PRAM method. We evaluate PRAM in three MDP environments: ReFrozenLake with 64 states and 4 actions, ConTaxi with 500 states and 6 actions, and Stochastic Battery Management (SBM) with 1000 states and 5 actions. Detailed descriptions of these environments are provided in Appendix B.1.

Distribution preservation. We first verify whether PRAM preserves the action distribution for every state and every message. To this end, we quantify the total variation (TV) distance between the reference policy π and the empirical action distribution induced by PRAM under each message $m \in \mathcal{M}$. Specifically, we define $D_{\mathrm{TV}}(m) \triangleq \frac{1}{2} \sum_s \mu(s) \sum_a |\pi(a|s) - P(a|s, m)|$, where $\mu(\cdot)$ denotes the stationary state distribution induced by the reference policy, and $P(a|s, m)$ is the empirical probability of selecting action a in state s when message m is transmitted using PRAM. We set the message length to $B = 3$ bits and blocklength to $n = 20$, and evaluate $D_{\mathrm{TV}}(m)$ for all eight possible messages. The results are reported in Table 1. The TV distance remains close to zero across all



(a) ReFrozenLake (b) ConTaxi (c) SBM

Figure 3: PRAM performance on policy watermarking in ReFrozenLake and SBM, and trajectory provenance in ConTaxi.

tested messages and environments, empirically confirming the per-message distribution-preserving property of PRAM. Consequently, PRAM can embed information without altering the nominal action distribution and hence without degrading the expected control reward. Additional results in Appendix B.2 further show that the control reward obtained under each transmitted message closely matches that of the reference policy.

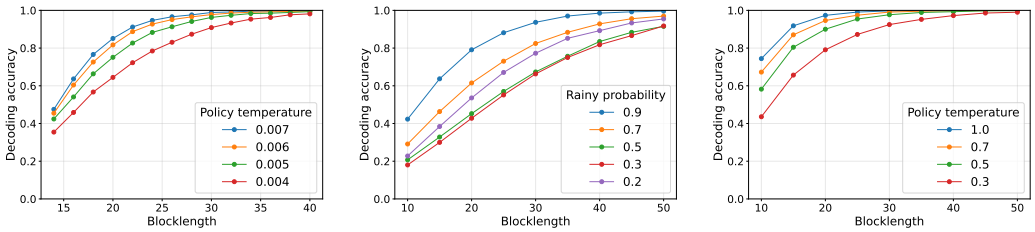
Applications. The ability of PRAM to embed information into stochastic policy realizations while preserving the prescribed action distribution naturally enables several downstream applications. We consider two representative tasks: *multi-bit policy watermarking* and *trajectory provenance*. In policy watermarking, a fixed multi-bit identifier is associated with a policy and repeatedly embedded into its behavior, providing a persistent signature for ownership verification. In contrast, trajectory provenance aims to determine the origin of an observed trajectory, e.g., which deployed agent or execution session produced it; the embedded information may therefore vary across episodes, agents, or deployment instances. We evaluate multi-bit policy watermarking in ReFrozenLake and SBM, and trajectory provenance in ConTaxi. For each task, we vary the message length and report the decoding accuracy as a function of the trajectory blocklength.

Although these two applications differ in the semantics of the embedded information, both can be realized through the same PRAM communication mechanism. For the watermarking task, we report performance averaged over different identifiers, making the experimental protocol comparable to that used for trajectory provenance. The results are shown in Figure 3. The decoding accuracy is defined as the fraction of trials in which all message bits are correctly recovered, i.e., $1 - P_e^{(n)}$. As shown in the figure, the decoding accuracy increases monotonically with the blocklength and approaches one for all tested message lengths even at reasonably low trajectory lengths ($\sim$ 40-50).

Effect of randomness. We next investigate how different sources of randomness affect the communication performance of PRAM. In particular, we consider randomness arising from both the stochastic reference policy and the system dynamics. To study the effect of policy randomness, we obtain the reference policy using entropy-regularized value iteration with temperature parameter α, where $\alpha = 0$ corresponds to the optimal deterministic policy and a larger α induces a more stochastic policy. We then evaluate PRAM under reference policies with different temperatures in ReFrozenLake and SBM. The results are shown in Figures 4a and 4c. As expected, greater policy randomness generally leads to higher decoding accuracy, since it provides more degrees of freedom for embedding information.

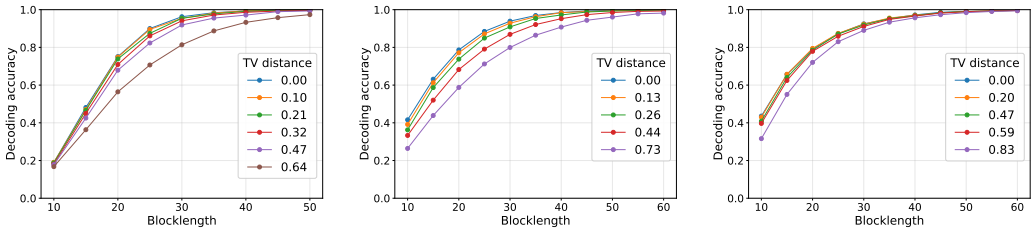
We examine the effect of system randomness in ConTaxi, where the taxi moves in the intended direction with probability p and in either lateral direction with probability $(1-p)/2$. Hence the resulting transition uncertainty is characterized by the entropy $H(p) = -p \log p - (1-p) \log \frac{1-p}{2}$. We fix the reference policy and evaluate PRAM in ConTaxi under different rainy probabilities p, as shown in Figure 4b. From a communication perspective, randomness in the system dynamics acts as channel noise and makes the transmitted information more difficult to recover from the state trajectory. Therefore, lower transition uncertainty generally leads to higher decoding accuracy.

Model mismatch. We further examine the robustness of PRAM to model mismatch. Specifically, suppose that neither the agent nor the observer knows the true transition kernel T of the MDP, and both instead rely on an estimated kernel $\hat{T} \neq T$. PRAM therefore performs action selection



(a) ReFrozenLake (b) ConTaxi (c) SBM

Figure 4: Effect of randomness on PRAM. (a) and (c): Policy randomness is varied through the temperature parameter. (b) System randomness is varied through the rainy probability.



(a) ReFrozenLake (b) ConTaxi (c) SBM

Figure 5: Performance of PRAM under model mismatch. The TV distance measures the discrepancy between the true transition kernel and the estimated kernel used by PRAM.

and posterior updates using $\hat{T}$, while the observed state trajectory is generated according to the true kernel T. Such model mismatch naturally arises in RL, where the transition dynamics must typically be estimated from interaction data.

We quantify the discrepancy between T and $\hat{T}$ using the occupancy-weighted TV distance $d(T, \hat{T}) \triangleq \frac{1}{2} \sum_s \mu(s) \sum_a \pi(a|s) \sum_{s'} |T(s'|s,a) - \hat{T}(s'|s,a)|$, where μ is the stationary state distribution induced by the reference policy π. Figure 5 reports the decoding performance of PRAM under different levels of model mismatch. For each environment, we estimate $\hat{T}$ from interaction trajectories, with larger amounts of data yielding more accurate estimates and hence smaller TV distances. We then fix the message length and evaluate the decoding accuracy as a function of the blocklength. The results demonstrate that PRAM is robust to moderate model mismatch: high decoding accuracy can still be achieved even when $\hat{T}$ differs noticeably from the true transition kernel. Moreover, across the tested mismatch levels, the decoding accuracy continues to improve with the blocklength and approaches one for sufficiently long trajectories.

6 Conclusion

This paper studied how to embed information into an agent's behavior without altering the action distribution prescribed by a stochastic control policy. We introduced the idea of *randomness modulation* and formulated the problem as DPIC. The key idea is to view the MDP environment as a communication channel, with the agent's action serving as the channel input and the resulting state as the channel output. Unlike conventional channel coding, DPIC imposes a stronger per-message distribution-preserving requirement. To address this challenge, we introduced common randomness shared by the encoder and decoder and developed the PRAM scheme. By modulating action realizations, PRAM establishes message-dependent couplings between the action and the common random variable, enabling information recovery from the observed state trajectory while preserving the prescribed action distribution. We proved the per-message distribution-preserving property of PRAM and established its asymptotic reliability. Extensive experiments demonstrated reliable message recovery, robustness to model mismatch, and the applicability of PRAM to tasks including multi-bit policy watermarking and trajectory provenance. Extending DPIC to partially observable settings, where the observer has only incomplete or noisy state information (Krishnamurthy, 2016), is an important and challenging direction for future work.

AI USE STATEMENT

The authors acknowledge the use of ChatGPT to assist with specific aspects of this research, as detailed below.

- **Research Design / Formulation (e.g., Proofs / Synthetic Data):** No AI tools were used for the core mathematical formulation or synthetic data generation.
- **Methodology and Experiments:** No AI tools were used to develop the methodology or design the experiments.
- **Writing, Editing, and Coding Assistance:** The authors drafted the manuscript independently and subsequently used ChatGPT to improve the clarity and presentation of the text. In addition, ChatGPT was used to assist with implementing the Python code for the experiments. The authors reviewed and verified the correctness of the implementation.

The authors have reviewed and verified all AI-assisted outputs and take full responsibility for the integrity and accuracy of this work.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the details of our method in Section 4 and the experimental setup in Section 5 and Appendix B.1. The algorithm pseudocode is provided in Appendix C. Complete proofs of the theoretical results are provided in Appendix A. Code and instructions for reproducing the experiments are included in the supplementary material.

REFERENCES

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pp. 151–160. PMLR, 2019.

Michael Amir, Manon Flageat, and Amanda Prorok. Remotely detectable robot policy watermarking. In *International Conference on Learning Representations*, volume 2026, pp. 149601–149623, 2026.

Hyeseon An, Shinwoo Park, Dongsu Kim, and Yo-Sub Han. Sequential behavioral watermarking for llm agents. *arXiv preprint arXiv:2605.11036*, 2026.

Vahid Behzadan and William Hsu. Sequential triggers for watermarking of deep reinforcement learning policies. *arXiv preprint arXiv:1906.01126*, 2019.

Gongpu Chen and Deniz Gündüz. Implicit communication in linear quadratic Gaussian control systems. *arXiv preprint arXiv:2509.16146*, 2025.

Gongpu Chen and Deniz Gündüz. The control plant as a communication channel: Implicit communication for decentralized lqg control. *arXiv preprint arXiv:2605.01903*, 2026.

Kangjie Chen, Shangwei Guo, Tianwei Zhang, Shuxin Li, and Yang Liu. Temporal watermarks for deep reinforcement learning models. In *Proceedings of the 20th international conference on autonomous agents and multiagent systems*, pp. 314–322, 2021.

Thomas M Cover, Joy A Thomas, and John Kieffer. *Elements of information theory*, volume 2. wiley New York, 1991.

Benjamin Eysenbach and Sergey Levine. Maximum entropy RL (provably) solves some robust rl problems. *arXiv preprint arXiv:2103.06257*, 2021.

Robert G. Gallager and Bariş Nakiboğlu. Variations on a theme by schalkwijk and kailath. *IEEE Transactions on Information Theory*, 56(1):6–17, 2010. doi: 10.1109/TIT.2009.2034896.

Zheng Gao, Xiaoyu Li, Xiaoyan Feng, Jiaojiao Jiang, Yang Song, Yulei Sui, Zhenchang Xing, and Liming Zhu. Trace: A two-channel robust attribution watermark via complementary embeddings for llm-agent trajectories. *arXiv preprint arXiv:2607.08400*, 2026.

Peter Hall, Christopher C Heyde, ZW Birnbaum, and E Lukacs. *Martingale limit theory and its application*, volume 12. Elsevier, 1980.

Kaibo Huang, Zipei Zhang, Zhongliang Yang, and Linna Zhou. Agent guide: A simple agent behavioral watermarking framework. *arXiv preprint arXiv:2504.05871*, 2025.

Mustafa O. Karabag, Melkior Ornik, and Ufuk Topcu. Least inferable policies for markov decision processes. In *2019 American Control Conference (ACC)*, pp. 1224–1231, 2019. doi: 10.23919/ACC.2019.8815129.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International conference on machine learning*, pp. 17061–17084. PMLR, 2023.

Vikram Krishnamurthy. *Partially observed Markov decision processes*. Cambridge university press, 2016.

Cheuk Ting Li and Abbas El Gamal. An efficient feedback coding scheme with low error probability for discrete memoryless channels. *IEEE Transactions on Information Theory*, 61(6):2953–2963, 2015.

Zifan Liu, Xinran Li, Shibo Chen, Gen Li, Jiashuo Jiang, and Jun Zhang. Reinforcement learning with intrinsically motivated feedback graph for lost-sales inventory control. In *International Conference on Artificial Intelligence and Statistics*, pp. 1657–1665. PMLR, 2025.

Thanh Thi Nguyen and Vijay Janapa Reddi. Deep reinforcement learning for cyber security. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):3779–3795, 2021.

Shinwoo Park, Hyejin Park, Hyeseon Ahn, and Yo-Sub Han. WaterMod: Modular token-rank partitioning for probability-balanced LLM watermarking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32683–32691, 2026.

Haim Henry Permuter, Tsachy Weissman, and Andrea J Goldsmith. Finite state channels with time-invariant deterministic feedback. *IEEE Transactions on Information Theory*, 55(2):644–662, 2009.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Aharon Rips and Oron Sabag. Communication over lqg control systems: A convex optimization approach to capacity. *arXiv preprint arXiv:2509.17002*, 2025.

Ofer Shayevitz and Meir Feder. Optimal feedback communication via posterior matching. *IEEE Transactions on Information Theory*, 57(3):1186–1222, 2011.

Samuel Sokota, Christian A Schroeder De Witt, Maximilian Igl, Luisa M Zintgraf, Philip Torr, Martin Strohmeier, Zico Kolter, Shimon Whiteson, and Jakob Foerster. Communicating via markov decision processes. In *International Conference on Machine Learning*, pp. 20314–20328. PMLR, 2022.

Sekhar Tatikonda and Sanjoy Mitter. The capacity of channels with feedback. *IEEE Transactions on Information Theory*, 55(1):323–349, 2008.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. Towards codable watermarking for injecting multi-bits information to llms. In *International Conference on Learning Representations*, volume 2024, pp. 39375–39401, 2024.

Haotian Wu, Gongpu Chen, and Deniz Gunduz. Actions speak louder than words: Rate-reward trade-off in Markov decision processes. In *International Conference on Learning Representations*, volume 2025, pp. 37518–37545, 2025.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. Advancing beyond identification: Multi-bit watermark for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4031–4055, 2024.

Liang Yu, Shuqi Qin, Meng Zhang, Chao Shen, Tao Jiang, and Xiaohong Guan. A review of deep reinforcement learning for smart building energy management. *IEEE Internet of Things Journal*, 8(15):12046–12063, 2021.

Rui Zhao, Xudong Sun, and Volker Tresp. Maximum entropy-regularized multi-goal reinforcement learning. In *International Conference on Machine Learning*, pp. 7553–7562. PMLR, 2019.

Junlin Zhu, Baizhou Huang, and Xiaojun Wan. Quantilemark: A message-symmetric multi-bit watermark for llms. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6790–6806, 2026.

APPENDIX

A TECHNICAL PROOFS

A.1 PROOF OF THEOREM 1

Proof. The desired result follows from an argument analogous to that used in Lemma II.1 of (Shayevitz & Feder, 2011). In particular, for every $m \in \mathcal{M}$,

$$\begin{aligned} P(F^{-1}_{\mathbf{a}_t|\mathbf{s}_t}(\mathbf{w}_t(m)) \le a) &= P\left(\inf\{a' : F_{\mathbf{a}_t|\mathbf{s}_t}(a') \ge \mathbf{w}_t(m)\} \le a\right) \\ &\overset{(a)}{=} P\left(F_{\mathbf{a}_t|\mathbf{s}_t}(a) \ge \mathbf{w}_t(m)\right) \\ &\overset{(b)}{=} F_{\mathbf{a}_t|\mathbf{s}_t}(a). \end{aligned}$$

Here, (a) holds because a CDF is non-decreasing and continuous from the right, and (b) follows from the fact that $\mathbf{w}_t(m) \sim \mathrm{Unif}[0,1]$ for every $m \in \mathcal{M}$. By definition, the above implies that $\mathbf{a}_t = F^{-1}_{\mathbf{a}_t|\mathbf{s}_t}(\mathbf{w}_t(m)) \sim \pi(\cdot|\mathbf{s}_t)$ for every realization m of the message. □

A.2 PROOF OF THEOREM 2

This part provides the proof of Theorem 2. Before that, we need the following lemma.

Lemma 1. *Let $m^* \in \mathcal{M}$ be the message transmitted by the agent. For any $m \neq m^*$, define*

$$K_t(m) = \mathbf{1}\{\mathbf{s}_t = s^*, \mathbf{a}_t(m^*) \neq \mathbf{a}_t(m)\}.$$

Then we have

$$\liminf_{n\to\infty} \frac{1}{n} K_n(m) \ge \delta_m,$$

where $\delta_m > 0$ is a constant.

Proof. Define the message point before applying random cyclic shift as

$$w'_t(m) = b_t(m) + \frac{1}{2}\rho_t(m).$$

Let

$$\alpha_t(m) \triangleq \min\{|w'_t(m^*) - w'_t(m)|, 1 - |w'_t(m^*) - w'_t(m)|\}$$

be the circular distance between $w'_t(m^*)$ and $w'_t(m)$. Then for any time t, we have $\alpha_t(m) \le 0.5$ and

$$\alpha_t(m) \ge \frac{\rho_t(m^*) + \rho_t(m)}{2} \ge \frac{\rho_t(m^*)}{2}.$$

For any $m \neq m^*$, define the likelihood ratio

$$R_t(m) = \frac{P(\mathbf{s}^t, \mathbf{u}^{t-1}|m)}{P(\mathbf{s}^t, \mathbf{u}^{t-1}|m^*)}$$

According to Bayes' rule, we have

$$\begin{aligned} \rho_t(m^*) &= \frac{P(\mathbf{s}^t, \mathbf{u}^{t-1}, m^*)}{P(\mathbf{s}^t, \mathbf{u}^{t-1})} = \frac{P_{\mathbf{m}}(m^*)P(\mathbf{s}^t, \mathbf{u}^{t-1}|m^*)}{\sum_{m\in\mathcal{M}} P_{\mathbf{m}}(m)P(\mathbf{s}^t, \mathbf{u}^{t-1}|m)} \\ &= \frac{P_{\mathbf{m}}(m^*)}{P_{\mathbf{m}}(m^*) + \sum_{m\neq m^*} P_{\mathbf{m}}(m)R_t(m)}. \end{aligned}$$

We next establish a lower bound on $\rho_t(m^*)$ by analyzing $R_t(m)$. By definition,

$$R_{t+1}(m) = R_t(m)\frac{T(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t(m))}{T(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t(m^*))}.$$

Then we can verify that $R_t(m)$ is a nonnegative martingale. To see this, let $\mathcal{G}_t = \sigma(\mathbf{s}^t, \mathbf{u}^{t-1})$ denote the sigma-algebra generated by $(\mathbf{s}^t, \mathbf{u}^{t-1})$. Then

$$\begin{aligned}\mathbb{E}[R_{t+1}(m)|\mathcal{G}_t] &= R_t(m) \sum_{s\in\mathcal{S}} T(\mathbf{s}_{t+1} = s|\mathbf{s}_t, \mathbf{a}_t(m^*)) \frac{T(\mathbf{s}_{t+1} = s|\mathbf{s}_t, \mathbf{a}_t(m))}{T(\mathbf{s}_{t+1} = s|\mathbf{s}_t, \mathbf{a}_t(m^*))} \\ &= R_t(m).\end{aligned}$$

Applying the martingale convergence theorem yields

$$R_t(m) \to R_\infty(m) < \infty \qquad \text{a.s.}$$

Therefore, there exists constants $C_m > 0$ and $N_m > 1$ such that

$$\alpha_t(m) \geq C_m, \quad \forall t \geq N_m. \tag{2}$$

At the state s^* specified in Assumption 2, define

$$f(u) = F^{-1}_{\mathbf{a}|s^*}(u)$$

Recall that

$$\mathbf{a}_t(m) = f(\mathbf{u}_t + w'_t(m)) \mod 1$$

Since $\mathbf{u}_t \sim \text{Unif}[0, 1]$, we can shift coordinates and define

$$q(\alpha) \triangleq P_{\mathbf{u}}(f(\mathbf{u}) \neq f(\mathbf{u} + \alpha)).$$

Here, $q(\alpha)$ is the probability that $\mathbf{a}_t(m^*) \neq \mathbf{a}_t(m)$ given that the circular distance between the two message points is α. Since $\pi(\cdot|s^*)$ is non-deterministic, we have $q(\alpha) > 0$ for every $\alpha \in (0, 1)$. Moreover, $q(\alpha)$ is continuous in α and

$$\alpha_t(m) \in [C_m, 0.5],$$

compactness gives

$$q_m \triangleq \inf_{\alpha\in[C_m,0.5]} q(\alpha) > 0.$$

Therefore,

$$P(\mathbf{a}_t(m^*) \neq \mathbf{a}_t(m)|\mathbf{s}_t = s^*) \geq q_m > 0 \tag{3}$$

for sufficiently large t. Define the state visitation frequency under the coding scheme as

$$\theta_n(s) = \frac{1}{n} \sum_{t=1}^{n} P(\mathbf{s}_t = s).$$

Then Assumption 1 guarantees that

$$\theta(s^*) \triangleq \lim_{n\to\infty} \theta_n(s^*) > 0. \tag{4}$$

Now,

$$\mathbb{E}[K_t(m)|\mathcal{G}_t] = \mathbf{1}\{\mathbf{s}_t = s^*\} q(\alpha_t(m))$$

For sufficiently large t, we have $\mathbb{E}[K_t(m)|\mathcal{G}_t] \geq q_m \mathbf{1}\{\mathbf{s}_t = s^*\}$. Now define the martingale difference

$$Z_t(m) = K_t(m) - \mathbb{E}[K_t(m)|\mathcal{G}_t].$$

Since $K_t(m) \in \{0, 1\}$, $\mathbb{E}[K_t(m)|\mathcal{G}_t] \in [0, 1]$. We thus have $|Z_t(m)| < 1$. The martingale strong law (Hall et al., 1980) gives

$$\frac{1}{n} \sum_{t=1}^{n} Z_t(m) \to 0 \quad \text{a.s.}$$

This implies that

$$\liminf_{n\to\infty} \frac{1}{n} K_n(m) = \liminf_{n\to\infty} \frac{1}{n} \mathbb{E}[K_n(m)] \geq q_m \theta(s^*) > 0.$$

This completes the proof.

□

Now we are ready to establish Theorem 2.

Proof of Theorem 2. Fix an arbitrary message $m \neq m^*$ and define the log-posterior ratio

$$L_t(m) = \log \frac{\rho_t(m^*)}{\rho_t(m)},\ t \geq 1.$$

By definition, $\rho_1(m) = P_{\mathbf{m}}(m) > 0$ for every $m \in \mathcal{M}$; hence, $L_1(m) < \infty$ for every m. Because $L_t(m)$ is determined by $(\mathbf{u}^{t-1}, \mathbf{s}^t)$, it can be viewed as a random variable. Define the stopping time

$$\tau_m \triangleq \inf\{t : \rho_t(m) = 0\}.$$

To establish the desired result, it suffices to show that, almost surely, either (i) $\tau_m < \infty$; or (ii) $L_t(m) \to \infty$ as $t \to \infty$. Indeed, the posterior update rule in (1) implies that once $\rho_i(m) = 0$, it remains zero for all $t \geq i$. Thus, under the convention $\log(x/0) = \infty$, case (i) also yields $L_t(m) \to \infty$.

Because $\rho_1(m) > 0$, case (i) can occur only under the following condition:

C1. There exists a tuple $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ such that $T(s' \mid s, a) = 0$.

We consider the two settings separately: first, we analyze case (i) when Condition **C1** holds; we then establish case (ii) when Condition **C1** does not hold.

Suppose that Condition **C1** holds. Because $\mathbf{a}_t(m)$ is generated using the uniform common randomness $\mathbf{u}_t$, see Figure 2 for an illustration, it can take any action in the support of $\pi(\cdot|\mathbf{s}_t)$ with positive probability. Consequently, when the process visits state s, there is a positive probability that $\mathbf{a}_t(m) = a$ while the transition generated under the true message m^* yields $\mathbf{s}_{t+1} = s'$. Since $T(s'|s, a) = 0$, the observed transition is impossible under the candidate message m, and the posterior update therefore gives $\rho_{t+1}(m) = 0$. Under the assumption that the Markov chain induced by the reference policy π is recurrent, this elimination event occurs within a finite time almost surely, implying that $\tau_m < \infty$ almost surely.

We next show that case (ii) is true when Condition **C1** does not hold. Define

$$\beta_t(m) = \log \frac{T(\mathbf{s}_{t+1}|\mathbf{a}_t(m^*), \mathbf{s}_t)}{T(\mathbf{s}_{t+1}|\mathbf{a}_t(m), \mathbf{s}_t)},\ m \in \mathcal{M}.$$

Then we have

$$L_{t+1}(m) = L_t(m) + \beta_t(m).$$

As a result,

$$L_n(m) = L_1(m) + \sum_{t=1}^{n-1} \beta_t(m)$$

Let $\mathcal{G}_t := \sigma(\mathbf{s}^{t+1}, \mathbf{u}^t)$ denote the sigma-algebra generated by $\mathbf{s}^{t+1}$ and $\mathbf{u}^t$. Define the conditional mean

$$d_t(m) = \mathbb{E}[\beta_t|\mathcal{G}_t].$$

Since $\mathbf{s}_{t+1} \sim T(\cdot|\mathbf{s}_t, \mathbf{a}_t(m^*))$,

$$\begin{aligned} d_t(m) &= \sum_s T(\mathbf{s}_{t+1} = s|\mathbf{s}_t, \mathbf{a}_t(m^*)) \log \frac{T(\mathbf{s}_{t+1} = s|\mathbf{a}_t(m^*), \mathbf{s}_t)}{T(\mathbf{s}_{t+1} = s|\mathbf{a}_t(m), \mathbf{s}_t)} \\ &= D_{\mathrm{KL}}\left(T(\cdot|\mathbf{s}_t, \mathbf{a}_t(m^*))||T(\cdot|\mathbf{s}_t, \mathbf{a}_t(m))\right) \\ &\geq 0. \end{aligned}$$

Now, let

$$\varphi_t(m) = \beta_t(m) - d_t(m).$$

Then by construction,

$$\mathbb{E}[\varphi_t(m)|\mathcal{G}_t] = 0.$$

Then

$$\phi_i(m) = \sum_{t=1}^{i-1} \varphi_t(m).$$

is a martingale with respect to the natural filtration. Specifically, $\phi_1(m) = 0$. We can decompose $L_n(m)$ as follows:

$$L_n(m) = L_1(m) + \sum_{t=1}^{n-1} d_t(m) + \phi_n(m). \tag{5}$$

When Condition **C1** does not hold, we have

$$|\beta_t(m)| \leq C$$

for some positive constant C. It follows that

$$0 \leq d_t(m) = \mathbb{E}[\beta_t|\mathcal{G}_t] \leq C.$$

Applying the triangle inequality yields

$$|\varphi_t(m)| = |\beta_t(m) - d_t(m)| \leq 2C.$$

Hence the martingale $\phi_n(m)$ has bounded increments. The Azuma-Hoeffding inequality then gives that, for any $\epsilon > 0$,

$$P\left(|\phi_n(m) - \phi_1(m)| \geq \epsilon n\right) \leq 2\exp\left(-\frac{\epsilon^2 n}{8C^2}\right).$$

Since $\phi_1(m) = 0$ and the RHS tends to zero as $n \to \infty$, we obtain

$$\sum_{n=1}^{\infty} P\left(|\phi_n(m)| \geq \epsilon n\right) < \infty.$$

Then, by Borel-Cantelli lemma, we know that the event

$$|\phi_n(m)| \geq \epsilon n$$

occurs only finitely often almost surely. Since this holds for any $\epsilon > 0$, we have

$$\lim_{n\to\infty} \frac{|\phi_n(m)|}{n} = 0,$$

almost surely.

We next focus on the second term of equation (5). By definition,

$$\begin{aligned}
\sum_{t=1}^{n-1} d_t(m) &= \sum_{t=1}^{n-1} D_{\text{KL}}\left(T(\cdot|\mathbf{s}_t, \mathbf{a}_t(m^*))||T(\cdot|\mathbf{s}_t, \mathbf{a}_t m)\right) \\
&\geq \sum_{t=1}^{n-1} \mathbf{1}\{\mathbf{s}_t = s^*, \mathbf{a}_t(m^*) \neq \mathbf{a}_t(m)\} D_{\text{KL}}\left(T(\cdot|\mathbf{s}_t, \mathbf{a}_t(m^*))||T(\cdot|\mathbf{s}_t, \mathbf{a}_t m)\right) \\
&\geq d_{\min} \sum_{t=1}^{n-1} \mathbf{1}\{\mathbf{s}_t = s^*, \mathbf{a}_t(m^*) \neq \mathbf{a}_t(m)\},
\end{aligned}$$

where

$$d_{\min} = \min_{a,a':a\neq a'} D_{\text{KL}}\left(T(\cdot|s^*, a)||T(\cdot|s^*, a')\right).$$

Combining the above arguments yields

$$\begin{aligned}\lim_{n\to\infty}\frac{1}{n}L_n(m) &= \lim_{n\to\infty}\left[\frac{L_1(m)}{n}+\frac{1}{n}\sum_{t=1}^{n-1}d_t(m)+\frac{\phi_n(m)}{n}\right]\\ &= \lim_{n\to\infty}\frac{1}{n}\sum_{t=1}^{n-1}d_t(m)\\ &\geq \delta_m > 0.\end{aligned}$$

The last line follows from Lemma 1. As a result, almost surely,

$$L_n(m) = \log\frac{\rho_n(m^*)}{\rho_n(m)} \to \infty \quad \text{as } n\to\infty.$$

Since the above statement holds for all $m \neq m^*$, we conclude that

$$\sum_{m\neq m^*}\rho_n(m) \xrightarrow{\text{a.s.}} 0, \rho_n(m^*) \xrightarrow{\text{a.s.}} 1, \quad \text{as } n\to\infty.$$

This completes the proof. □

B Experiment Details

B.1 Experimental environments

This section provides brief descriptions of the experimental environments used to evaluate the proposed PRAM method. Detailed Python implementations of these environments are included in the supplementary material and will be open-sourced after the review process.

ReFrozenLake. The Revised FrozenLake (ReFrozenLake) is a variant of FrozenLake in Gymnasium, modified to provide a continuing finite-state MDP suitable for long-horizon communication experiments. The environment retains the original grid layout and slippery transition dynamics, but removes terminal states before reaching the goal. In particular, reaching a hole no longer terminates the episode; instead, the agent incurs a penalty and is randomly transitioned to another valid position in the grid. Reaching the goal yields a positive reward and terminates the trajectory.

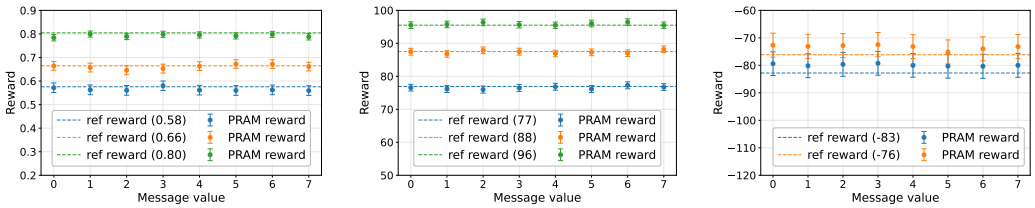
ConTaxi. The Continuing Taxi (ConTaxi) is a variant of Taxi-v4 in Gymnasium that transforms the original episodic task into a continuing environment. It preserves the original state and action spaces, including the taxi's location, passenger location, destination, and six available actions. When the taxi successfully drops off a passenger, the episode does not terminate; instead, the taxi remains at the drop-off location and a new passenger origin and destination are sampled uniformly from all valid pairs. The environment therefore supports an uninterrupted sequence of trips while retaining Taxi-v4's reward structure and optional rainy-mode dynamics.

SBM. Stochastic Battery Management (SBM) is a customized finite-state, finite-action energy-management environment designed to model stochastic battery charging and discharging. The state is $s = (b, d, p)$, where $b \in \{0, \ldots, 19\}$ denotes the battery level, $d \in \{-4, \ldots, 5\}$ denotes the net electricity demand, and $p \in \{1, \ldots, 5\}$ denotes the electricity-price level, resulting in 1000 states in total. The action space is $\mathcal{A} = \{-2, -1, 0, 1, 2\}$, corresponding to different charging and discharging rates.

The commanded battery action is executed with stochastic perturbation: with probability 0.8 it is executed as intended, while with probability 0.1 it is shifted by -1 or $+1$, respectively, subject to the action and battery-capacity constraints. The net demand and electricity price independently evolve according to nearest-neighbor Markov chains. The overall transition kernel therefore factorizes into the battery, demand, and price transition probabilities.

The reward reflects the electricity purchasing cost, revenue from selling excess energy, and battery degradation cost. Let $x = b' - b$ denote the realized battery change and $g = d + x$ the resulting grid electricity exchange. The instantaneous reward is

$$r = -p[g]^+ + \beta p[-g]^+ - \lambda|x|,$$



(a) ReFrozenLake (b) ConTaxi (c) SBM

Figure 6: Control reward of transmitting different messages using PRAM.

where $[z]^+ = \max\{z, 0\}$, $\beta = 0.5$ is the selling-price ratio, and $\lambda = 0.1$ controls the battery degradation penalty. The environment is continuing and contains no terminal states.

B.2 Additional Experimental Results

In DPIC, the per-message distribution-preserving requirement is motivated by the need to maintain control performance while embedding watermarks or other information into the agent's behavior. Table 1 has shown that the empirical action distributions induced by PRAM closely match the corresponding reference policies. Here, we further verify that PRAM indeed preserves the control reward. Specifically, in each of the three environments, we select two or three reference policies and evaluate their average rewards, shown as dashed lines in the figure. For each reference policy, we then evaluate the reward achieved by PRAM when transmitting different messages. As shown in the figure, the rewards under PRAM are nearly identical across messages and closely match those of the corresponding reference policies.

C Algorithm Pseudocode

The encoding procedure of PRAM at the agent is summarized in Algorithm 1.

Algorithm 1 Posterior-matching randomness modulation (PRAM): Encoder

Require: Transition kernel T, blocklength n, message $m^* \in \mathcal{M}$, message prior distribution $P_{\mathbf{m}}$

1: **Initialization:** Let $\rho_1 = P_{\mathbf{m}}$. Observe the initial state s_1

2: **for** $t = 1$ to n **do**

3: Compute for every m:

$$b_t(m) \triangleq P(\mathbf{m} < m | \mathbf{s}^t, \mathbf{u}^{t-1}) = \sum_{j<m} \rho_t(j)$$

4: Draw a sample of the common random variable $\mathbf{u}_t \sim \mathrm{Unif}[0, 1]$

5: Determine the information point:

$$\mathbf{w}_t = [b_t(m^*) + l_t(m^*)\rho_t(m^*) + \mathbf{u}_t] \mod 1,$$

6: Compute the action

$$a_t = F^{-1}_{\mathbf{a}_t|\mathbf{s}_t}(\mathbf{w}_t) := \inf\{a \in \mathcal{A} : F_{\mathbf{a}_t|\mathbf{s}_t}(a) \geq \mathbf{w}_t\}$$

7: Apply action a_t and observe the next state s_{t+1}. Then update the posterior

$$\rho_{t+1}(m) = \frac{\rho_t(m)T(s_{t+1}|s_t, a_t(m))}{\sum_j \rho_t(j)T(s_{t+1}|s_t, a_t(j))}, \ \forall m \in \mathcal{M}$$

8: **end for**