Algorithms for Emergent Communication

Nivasini Ananthakrishnan, Mark Bedaywi, Michael I. Jordan, Stuart Russell, and Nika Haghtalab

University of California, Berkeley
{nivasini,mark_bedaywi,michael_jordan,russell,nika}@berkeley.edu

Abstract

What algorithmic principles drive the emergence of communication in multi-agent learning environments? While prior work on *emergent communication* has examined the properties and evolution of emergent languages, the fundamental question of which decentralized learning algorithms provably enable language emergence remains largely unexplored. To address this, we introduce an online variant of the Lewis signaling game, where a sender and receiver must encode and decode a sequence of generated states cooperatively. The emergence of a shared language is captured by a notion of *communication regret* in this environment — the regret experienced by the sender and receiver relative to the best synchronized encoder-decoder pair. We show that sender stability (i.e., low *switching regret*) and receiver adaptivity (i.e., low *tracking regret*) are sufficient conditions for emergent communication, achieving communication regret of $\mathcal{O}\left(T^{2/3}\right)$ and $\mathcal{O}\left(T^{1/2}\right)$ when states are generated adversarially or stochastically, respectively. Beyond these sufficient conditions, we demonstrate that minimally tailoring these algorithms for synchronized multi-agent learning significantly accelerates communication, reducing adversarial communication regret to $\mathcal{O}\left(T^{1/2}\right)$.

1 Introduction

One of the most fascinating phenomena in multi-agent systems is the emergence of language among interacting agents, without any pre-specified language or conventions. While human language has evolved over millennia, advances in machine learning and AI now offer us a front-row seat for observing how language may emerge in multi-agent learning environments. This is studied across various disciplines including game theory, biology, linguistics, psychology, and computer science through the field of *emergent communication* [Steels, 1997, Smith and Harper, 2003, Lewis, 2008, Kirby et al., 2014, Lazaridou et al., 2016, Lazaridou and Baroni, 2020], which investigates how language and communication arise as a byproduct of decentralized training algorithms in cooperative multi-agent settings.

Much of the past work in this space has focused on the properties of languages that can emerge as byproducts of ad hoc choices of the training algorithms.

However, a key foundational question has remained largely unexplored:

> *What properties of the training algorithms provably lead to the emergence of communication? Do natural training algorithms exhibit these properties?*

Our approach. This is precisely the question we address in this work. To ground our approach in mathematical foundations, we introduce an *online* variant of the canonical Lewis signaling game [Lewis, 2008]—a widely used model of cooperative games used in the study of emergent communication. In this repeated cooperative game, a sender seeks to encode a state that is generated by nature (whether adversarially or stochastically) and a receiver seeks to decode this message in order to reconstruct the original state. Both

	Existing Generic Algorithms: Switching vs Tracking Regret	Specialized Algorithms for Synchronization	
		Initial Setup	Plain Mode
Stochastic Environment	$\tilde{\mathcal{O}}\left(T^{1/2}N^{1/2}M\right)$ (Cor. B.2)	$\tilde{\mathcal{O}}\left(T^{1/2}(\log N)^{1/2}M\right)$ (Cor. B.2)	$\tilde{\mathcal{O}}\left((T^{1/2}+N^3)M\right)$ (Cor. B.2)
Oblivious Adversarial Environment	$\tilde{\mathcal{O}}\left(T^{2/3}N^{1/3}M\right)$ (Thm. 5.3)	$\tilde{\mathcal{O}}\left(T^{1/2}(\log N)^{1/2}M\right)$ (Thm. 5.4)	$\tilde{\mathcal{O}}\left(T^{1/2}N^{3/2}M\right)$ (Prop. 5.5)
Adaptive Adversarial Environment	$\Omega(T)$ (Thm. A.2)	$\Omega(T)$ (Thm. A.2)	$\Omega(T)$ (Thm. A.2)

Table 1: An overview of our results on the communication regret of various protocols for when the state is generated adversarially or stochastically in the reconstruction game under various constraints on the ability of the agents to coordinate. N denotes the size of the state space, M denotes the size of the message space, and T denotes the time horizon. In the first column, we use generic algorithms that have no switching or tracking regret, without tailoring them for communication. In the second and third columns, we design algorithms that incorporate explicit synchronization mechanisms intended for improving communication. The "initial setup" refers to a protocol where the sender and receiver first establish a shared meaning for messages, while "plain mode" refers to a setting where meaning must emerge through online interactions alone. Note that in many settings, the dependence on T is $O(\sqrt{T})$, which is tight(Proposition E.1).

agents are rewarded for accurate reconstruction of the state. In this formalism, the *emergence of communication* is now captured by the *regret these agents incur relative to the best synchronized encoder-decoder pair of policies in hindsight. We call this their communication regret.*

Our goal is not to design algorithms that merely obtain optimal regret bounds, because unconstrained algorithm design may lead to highly engineered algorithms that do not make sense outside of the communication setting. In contrast, emergent communication studies how communication can arise organically through natural training algorithms that are used for far broader tasks. Still, not all training algorithms are equally conducive to efficient communication. Therefore, the right level of abstraction is to study algorithmic principles and identify broad algorithmic properties satisfied by many natural training algorithms that lead to low communication regret. This frames our goal as characterizing classes of learning algorithms—one for the sender and one for the receiver—such that when any algorithms from these classes are used by the sender and receiver, their communication regret grows sub-linearly with the number of communication rounds.

Once we identify these algorithmic properties and natural classes of algorithms demonstrating them, we may still minimally tailor these algorithms with the aim of sharpening their regret guarantees. This would offer a bridge between truly emergent communication and deliberate attempts at learning to communicate.

Our technical results. Our first set of results identifies sufficient conditions on sender's and receiver's algorithm for effective communication to emerge. The two key properties we identify are (1) sender stability, meaning the sender must avoid changing its encoding strategy too often while still maintaining low external regret, and (2) receiver adaptivity, meaning the receiver must achieve low external regret not only against a single decoding policy, but also against *sequences* of policies that minimally change over time. The former is formalized by the framework of *regret-minimization with switching costs* [Cesa-Bianchi et al., 2013] and the latter is formally known as *tracking regret minimization* [Herbster and Warmuth, 1998]. These two properties are sufficient to ensure a sub-linear regret to the best synchronized encoder-decoder pair of policies in hindsight. By designing pairs of computationally efficient no-regret algorithms that achieve optimal switching regret and optimal tracking regret, we show that a language emerges at a rate of $\mathcal{O}\left(T^{2/3}\right)$.

Our second set of results looks beyond stability and adaptivity as sufficient conditions. We show that tailoring these algorithms to more intentionally enable synchronization with the other agent can significantly

accelerate the emergence of communication. We provide a pair of computationally efficient algorithms that achieve a communication regret of $\mathcal{O}\left(T^{1/2}\right)$, which is tight due to standard regret lower bounds.

Beyond reconstruction as a goal of communication, we extend our model to a broader class of cooperative games with arbitrary utility. This broader class of utilities capture settings where the goal of communication is to achieve high rewards on downstream tasks rather than reconstruction of the original state. We prove that achieving sub-linear regret in these general settings is computationally intractable by reducing it to a maximum coverage problem. Nevertheless, we show that there exist pairs of computationally efficient algorithms that achieve sub-linear regret at a rate of $\mathcal{O}\left(T^{1/2}\right)$ relative to a $(1 - 1/e)$-fraction of the utility that the optimal encoder-decoder policy pair can achieve.

2 Model and Preliminaries

The online communication game is defined by a state space Ω of size N and a message space $\mathcal{M}$ of size M. It is a sequential game with incomplete information between a sender and a receiver, played repeatedly over T rounds. Nature initially fixes a sequence of state-generating distributions $(D_t)_{t=1}^{\infty}$. We call the special case when all distributions D_t are the same as the *stochastic setting*. Every round t of the communication game involves the following steps:

1. Nature draws a state $\omega_t^* \in \Omega$ from the state-generating distribution D_t.
2. The sender sees ω_t^* and sends a message $m_t \in \mathcal{M}$.
3. The receiver sees m_t from the sender but not ω_t^*. The receiver then decodes m_t to a state ω_t.
4. Both sender and receiver receive the reward $r(\omega_t^*, \omega_t) = \mathbb{1}\{\omega_t = \omega_t^*\}$.

In the normal form representation of the communication game, the sender's action space is Σ, which is the set of all encoding schemes $\sigma : \Omega \to \mathcal{M}$ that are (possibly randomized) mappings from states to messages. The receiver's action space is P which is the set of all decoding schemes $\rho : \mathcal{M} \to \Omega$ mapping messages to states.

The joint success of the sender and receiver in the online communication game is measured by the following notion of regret that we call the *communication regret*. We simply refer to this as regret in the remainder of the paper.

Definition 2.1 (Communication regret)**.** The communication regret of a sequence $\chi = (\omega_t^*, \sigma_t, \rho_t)_{t=1}^{T}$, written $R_T(\chi)$, is:

$$\max_{\sigma \in \Sigma, \rho \in \mathrm{P}} \sum_{t=1}^{T} \mathbb{1}\{\rho(\sigma(\omega_t^*)) = \omega_t^*\} - \sum_{t=1}^{T} \mathbb{1}\{\rho_t(\sigma_t(\omega_t^*)) = \omega_t^*\}.$$

We suppress ω_t^* in the regret notion when it's clear from the context. Moreover, we use $r_t(\sigma, \rho) = \mathbb{1}\{\rho(\sigma(\omega_t^*)) = \omega_t^*\}$ as a shorthand for the rewards of an encoder-decoder pair.

Remark 2.2 (Adaptive state-generating distribution are too hard)*.* In our work, we consider an *oblivious* nature whose choice of state-generating distributions can change over time, but does not depend on the actions of the sender and receiver. It turns out that if nature can *adaptively* choose strategies depending on the agents' actions, then there are games where it is impossible to achieve sub-linear communication regret as shown in Appendix A.

2.1 Background on Online Learning

As the notion of *communication regret* suggests, notions and algorithms originating from the online learning literature will play a central role in our paper.

In a setting with action set $[K]$ and T rounds, for sequences of actions and reward functions (a_t, r_t) where $a_t \in [K]$ and $r_t : [K] \to [0, 1]$, we can define the following notions of regret.

Definition 2.3 (External regret and number of switches)**.** The external regret of a sequence $(a_t, r_t)_{t=1}^T$ is

$$R_T = \max_{a \in [K]} \sum_{t=1}^{T} r_t(a) - \sum_{t=1}^{T} r_t(a_t).$$

The number of switches in the sequence is $\sum_{t=2}^{T} \mathbf{1}\{a_t \neq a_{t-1}\}$.

Definition 2.4 (Tracking regret with p segments)**.** Tracking regret [Herbster and Warmuth, 1998] measures a learner's ability to compete with a sequence of a p-times changing benchmark of actions rather than a single best fixed action. It is defined as:

$$R_T^{\text{track},p} = \max_{\substack{s_1 < \cdots < s_{p+1} \in [T] \\ s_1 = 1, s_{p+1} = T \\ b_1, \ldots, b_p \in [K]}} \sum_{i=1}^{p} \sum_{t=s_i}^{s_{i+1}} r_t(b_i) - \sum_{t=1}^{T} r_t(a_t).$$

This regret formulation is useful in changing environments where the optimal action varies over time.

Feedback models. Online learning algorithms select action a_t in a round t based on the available history consisting of actions and feedback of previous rounds. In a *full-information* setting the entire reward function r_t is revealed as feedback at the end of round t. In a *bandit* setting, only the reward of the selected action $r_t(a_t)$ is revealed as feedback.

3 Related Work

The Emergent Communication (EC) literature [Foerster et al., 2016, Lazaridou et al., 2016, Lazaridou and Baroni, 2020] studies how agents learn to communicate in Lewis signaling games when trained using standard training dynamics. This study is particularly important to building cooperative AI systems that understand and are aligned with human preferences and values [Dafoe et al., 2020, Hadfield-Menell et al., 2016, Shah et al., 2020].

One line of work studies the properties of the emergent language, particularly in terms of its efficiency and naturalness [Lowe et al., 2019]. Theoretical frameworks [Rita et al., 2022, Zion et al., 2024] have been built for this purpose.

Another line of research, more closely related to our work, investigates how the choice of training algorithms impacts performance. Empirical work has compared different choices of standard training algorithms and model architectures [Havrylov and Titov, 2017, Kim and Oh, 2021, Ren et al., 2019, Chaabouni et al., 2022, Rita et al., 2020] and the impact of having a population of agents [Guo et al., 2019, Graesser et al., 2019, Raviv et al., 2019a,b, Li and Bowling, 2019, Chaabouni et al., 2022].

Some of the empirical findings are adjacent to the insights about the advantages of having a stable sender and adaptive receiver from our work. Chaabouni et al. [2022] and Rita et al. [2020] show a benefit of stabilization for the sender through KL regularization and increasing the cost of communicating for the sender respectively. Li and Bowling [2019] show benefit of resetting of receivers, a form of adaptivity, that occurs during population interactions with new agents entering.

Training dynamics have received a more theoretical treatment in works in game theory and evolutionary biology [Franke, 2009b,a, Jäger, 2007, 2012, Trapa and Nowak, 2000, Kirby, 2002, Kirby et al., 2014, Jacob et al., 2023]. However, these works mostly view language formation as equilibrium computation. There are many possible equilibria and there is no guarantee of convergence to the optimal equilibrium, which is the goal of our work.

Computing the optimal equilibrium is computationally intractable [Gilboa and Zemel, 1989] in general, including in games of common interest [Chu and Halpern, 2001]. In common interest games, the optimal equilibrium is also each player's Stackelberg equilibrium. The computational tractability of computing the Stackelberg equilibrium is shown to depend on the geometry of the game [Letchford et al., 2009, Peng et al., 2019]. For games where the Stackelberg equilibrium can be computed efficiently, convergence to the

Stackelberg equilibrium can be achieved through dynamics where one player's learning dynamic is more stable than the other [Brown et al., 2024, Zrnic et al., 2021]. The learning dynamics we propose for communication games also satisfy this property.

4 Warm-Up: Compression With Automatic Synchronization

For a sender and receiver to communicate successfully in online communication games they have to overcome two challenges. The first is the problem of learning to optimally compress the state space into the size of the message space. The optimal compression depends on the sequence of states generated. We call this the *compression problem*. The second is to enable the sender and receiver to be *synchronized* with one another. That is, the sender's encoder is optimal given the receiver's decoder and the receiver's decoder is optimal given the sender's encoder and nature's strategy of the distribution over states.

As a warm-up, let us first study the compression problem in isolation by studying an idealized setting in which the sender and receiver are always synchronized, i.e. their actions are chosen in a centralized manner by one meta-player. Removing the key challenge of synchronization, this allow us to isolate and identify other factors which will play a role in the emergence of communication.

Definition 4.1 (Centralized communication game)**.** The centralized communication game is a repeated game between nature and a meta-player, where at every round players take actions simultaneously. [1] That is, nature chooses a state $\omega_t^* \in \Omega$ and the meta-player chooses an encoder-decoder pair (σ_t, ρ_t). Then the realized state ω_t^* is revealed to the meta-player and the meta-player receives a reward $r(\omega_t^*, \sigma_t, \rho_t) = \mathbb{1}\{\sigma_t(\rho_t(\omega_t^*)) = \omega_t^*\}$.

The main challenge in the centralized communication game is the large space of all encoder-decoder pairs to optimize over. However, there is still enough structure in the action space and utility functions to design computationally efficient algorithms with low regret as shown in the following proposition.

Proposition 4.2. *There is a* $\mathrm{poly}(M, N, T)$ *time algorithm for the meta-player in the centralized communication game (Definition 4.1), such that the meta-player's expected regret satisfies* $\mathbb{E}[R_T] \leq 2\sqrt{MT \log N}$.

Proof sketch. We will first reduce the meta-player's problem in the centralized communication game to a regret minimization problem with the following properties: 1) The action space is the combinatorial space of all M-sized subsets of $[N]$, 2) utilities per round are linear functions of the actions, 3) the offline problem of finding the best action given an arbitrary linear utility function can be solved computationally efficiently, and 4) all actions' utilities can be deduced after every round.

Using this reduction, we will use methods developed for online learning with such a combinatorial structured action space and linear utilities [Kalai and Vempala, 2005, Cesa-Bianchi and Lugosi, 2012] to efficiently obtain the regret bound stated in the proposition. Algorithms such as Follow the Perturbed Leader and Follow the Lazy Leader [Kalai and Vempala, 2005] will achieve the bounds in the theorem. The full proof is in Appendix F.1 Here is an outline of the arguments for the reduction.

The reduction: At round t, the meta-player chooses a deterministic decoder ρ_t and plays the pair $(\mathrm{BR}(\rho_t), \rho_t)$ in the centralized communication game. Here $\mathrm{BR}(\rho_t)$ is the best-response encoder corresponding to the decoder ρ_t. $\mathrm{BR}(\rho)$ maps any state ω to a message that has maximum likelihood (under ρ) of being decoded to ω. Note that $\mathrm{BR}(\rho_t)$ can be computed without knowledge of nature's strategy while the optimal encoder relative to a decoder depends on nature's strategy.

This form of restriction to the meta-player's algorithm is without any loss of utility. Clearly, there is no utility loss in always choosing the best-response decoder to every encoder. The restriction to deterministic decoders is also lossless since the optimal encoder-decoder pair in every round is deterministic as shown in Lemma F.1.

Now let us show that the reduction satisfies the properties we stated in the beginning.

[1] This means the meta-player chooses an action without having access to the realized state. This it to make the idealized setting serve as a building block for the decentralized setting where the receiver must choose a decoder without having access to the realized state.

1) Combinatorial structure of action space: The action space of the reduction is the space of all deterministic decoders, which we can equivalently represent as the space of M-sized subsets of $[N]$, where the subset corresponding to a deterministic decoder is the image set of the decoder i.e., states the decoder chooses at some message. We can represent ρ_t as a vector $\phi_t \in \{0,1\}^N$, where the coordinates indicate if state ω_i belongs to the image set of ρ_t.

2) Linear utilities: In the realized state ω_t^*, we can represent the utility of round t due to action $\phi_t \in \{0,1\}^N$ by $\phi_t \cdot \mathbb{1}_{\omega_t^*}$, where $\mathbb{1}_{\omega_t^*}$ is the one-hot encoding vector of ω_t^* with one in the coordinate corresponding to state ω_t^* and zero everywhere else. Hence the utilities are linear in the actions.

3) Efficiency of finding optimum action: For any utility vector $u \in \mathbb{R}^n$, the optimal deterministic decoder has a simple form that can be efficiently computed. In fact, the optimal decoder is one with value 1 in the M-coordinates of u with the largest values and zero everywhere else.

4) Information about utilities of all actions: Although the structure of the centralized game does not explicitly reveal the utilities of all actions, the meta-player can deduce the utility of every $(\mathrm{BR}(\rho), \rho)$ when the realized state ω_t^* is revealed. The utiltity is 1 if ω_t^* is in ρ's image set and 0 otherwise. □

5 Main Results

In our warmup study of centralized communication game, the strategies of the sender and receiver were always synchronized—i.e. the pair of encoder-decoders used in every round were optimal with respect to each others. However, in the original formulation of our game, the players have to choose encoding and decoding policies in a decentralized manner without knowing each other's chosen policy. In this case, the receiver's choice of decoder is no longer automatically synchronized with the sender's choice of encoder. Instead, the receiver needs to learn to adapt its decoder to be synchronized with the sender's encoder.

5.1 Switching and Tracking Regrets are Sufficient For Emergent Communication

In this section, we see how natural learning goals of external regret minimization with few switches (Definition 2.3) for the sender and tracking regret minimization (Definition 2.4) for the receiver are sufficient for the receiver to quickly catch up and adapt her decoding policy to the sender's encoding scheme, without any additional explicit effort towards synchronizing the two agents.

At a high level, every time the sender updates their encoding scheme, the receiver must spend some interaction rounds learning and adapting to the new scheme, during which both agents can incur an error. Intuitively, the fewer times the sender updates their policy (i.e., fewer switches), the easier it is for the receiver to catch up. However, the standard notion of external regret for the receiver is insufficient to ensure effective adaptation. Instead, competing with the moving benchmark defined by the sender's evolving policies is sufficient for the receiver to adapt and get synchronized with the sender. This is where tracking regret comes in handy: by ensuring that the receiver's actions are competitive with respect to any moving benchmark (with only a few segments), it ensures that the receiver's algorithm also effectively catches up to the sender's moving encoding scheme.

Below (Definition 5.2), we describe how the sender and receiver can use any generic algorithms $\mathcal{A}_{\text{stable}}$ and $\mathcal{A}_{\text{adaptive}}$, where $\mathcal{A}_{\text{stable}}$ is an online algorithm with full-information feedback for the actions space $\Sigma \times \mathrm{P}$ of encoder-decoder pairs and $\mathcal{A}_{\text{adaptive}}$ is an online learning with bandit-feedback over the action space of decoders P.

Remark 5.1 (Sender feedback). Note that the feedback structure of the communication game is not explicitly full-feedback for either player. However, since the sender sees the realized state, the sender can still compute the reward of every encoder-decoder pair, which is the probability that the realized state is reconstructed by the pair. Hence, it is possible for the sender to employ an algorithm $\mathcal{A}_{\text{stable}}$ over $\Sigma \times \mathrm{P}$ that relies on full-information feedback.

Definition 5.2 ($(\mathcal{A}_{\text{stable}}, \mathcal{A}_{\text{adaptive}})$-Stable sender, adaptive receiver protocol)**.** This protocol uses a stable regret minimizing algorithm $\mathcal{A}_{\text{stable}}$ and an adaptive regret minimizing algorithm $\mathcal{A}_{\text{adaptive}}$ in the following way:

Sender. The sender uses the algorithm $\mathcal{A}_{\text{stable}}$ to select encoder-decoder pairs $\left(\sigma_t^S, \rho_t^S\right)$ and employs the encoding strategy σ_t^S at round t. $\mathcal{A}_{\text{stable}}$ will typically output $\sigma_t^S = \text{BR}(\rho_t^S)$. Hence the sender choosing σ_t^S in this way enables the sender to optimize over the space of decoders instead of encoders, which is a smaller space.

Receiver. The receiver uses the algorithm $\mathcal{A}_{\text{adaptive}}$ to select the decoding strategy ρ_t^R for round t.

In the following theorem, we bound the communication regret of this protocol in terms of the external regret and number of switches of $\mathcal{A}_{\text{stable}}$ and tracking regret of $\mathcal{A}_{\text{adaptive}}$. The theorem also states how by using algorithms developed by works studying regret with switching costs and tracking regret as black boxes, we obtain a communication regret of $\mathcal{O}\left(T^{2/3}MN^{1/3}\right)$.

Theorem 5.3. *Suppose the sender and receiver follow the stable sender, adaptive receiver protocol (Definition 5.2) with the algorithms $\mathcal{A}_{\text{stable}}, \mathcal{A}_{\text{adaptive}}$, then the communication regret is at most*

$$R_T \leq R_T^{ext}(\mathcal{A}_{\text{stable}}) + R_T^{track}(\mathcal{A}_{\text{adaptive}}; S_T(\mathcal{A}_{\text{stable}}; \delta)) + \delta T,$$

for every $\delta > 0$, where $S_T(\mathcal{A}_{\text{stable}}; \delta)$ is the number of switches made by $\mathcal{A}_{\text{stable}}$ with probability at least $1 - \delta$ and R_T^{ext} and R_T^{track} denote external and tracking regrets.

There exist efficient algorithms $\mathcal{A}_{\text{stable}}, \mathcal{A}_{\text{adaptive}}$ that result in expected communication regret

$$\mathbb{E}[R_T] \in \mathcal{O}\left(MT^{2/3}(N \log N)^{1/3}(\log T)^{1/6} + M \log N\right).$$

5.2 Better bounds through algorithms tailored for communication

In the previous part, we saw communication protocols for the sender and receiver that used natural online learning algorithms as building blocks, without modifying them to specialize toward the communication problem. In this subsection, we will construct protocols that are more specialized and achieve better regret rates of $\mathcal{O}\left(T^{1/2}\right)$ in comparison to the rate of $\mathcal{O}\left(T^{2/3}\right)$ previously achieved. Note that the improved rate of $\mathcal{O}\left(T^{1/2}\right)$ achieves the optimal dependence on T as shown in Proposition E.1.

The protocols in this subsection improve the regret bound by reducing the cost to achieve synchronization after each switch by the sender. This is done by sending a sequence of messages to the receiver not about the current state, rather about the new policy the sender is committing to. In doing so, the receiver will not need to explore to best respond. This improved regret hence comes at the cost of requiring a more intricate coordination scheme. The following theorem describes a protocol that has a cost of $\mathcal{O}(M \log N)$ per switch.

Theorem 5.4. *(Synchronization with Initial Setup) Given any online algorithm $\mathcal{A}$ over the space $\Sigma \times \text{P}$ of encoder-decoder pairs, there are sender and receiver algorithms that achieve communication regret at most*

$$R_T \leq R_T^{ext}(\mathcal{A}) + M \log N \cdot S_T(\mathcal{A}; \delta) + \delta T,$$

for any $\delta > 0$, where $S_T(\mathcal{A}; \delta)$ is the number of switches made by $\mathcal{A}$ with probability at least $1 - \delta$.

There are efficient sender and receiver algorithms that result in expected regret

$$\mathbb{E}[R_T] \in \mathcal{O}\left(T^{1/2}M(\log N)^{1/2}\right).$$

Proof sketch. The sender uses $\mathcal{A}$ in mostly the same way as in the previous protocol (Definition 5.2), generating encoder-decoder pairs and employing the encoder of this pair in the communication game. However, instead of always employing the encoder from $\mathcal{A}$, the sender uses some rounds after switching encoders to explicitly communicate the new decoder the receiver should use to the receiver.

Since there are N^M deterministic decoders (all mappings from $[N]$ to $[M]$), the sender can communicate the new decoder to the receiver in $\log_M(N^M) = M \log_M(N)$ steps by assigning a unique sequence of messages to each decoder. This protocol crucially assumes that both the sender and receiver share a common mapping of sequence of messages to decoders.

Instead of communicating after every switch in encoder, the sender communicates after the first mistake made by the receiver after the switch. This is to ensure the receiver knows which rounds the sender is using to communicate the new decoder. A mistake after previous synchronization is a common signal for both the sender and receiver to start the meta-communication of communicating the new decoder.

After the completion of the meta-communication of the new decoder, the receiver exactly knows the decoder to use and becomes perfectly synchronized with the sender without any additional exploration. The cost of the synchronization is simply the $M \log_M(N)$ rounds used for the meta-communication.

We can get an even lower cost per switch than $M \log_M(N)$ via a more refined notion of stability which tracks the number of arguments that take different values after a switch. This is explored further in Appendix D.1. □

The protocol in the above proof relies on both the sender and the receiver having a shared meaning for messages before the start of the online interaction. That is, when the sender communicates a sequence of $M \log N$ messages after every shift, the sender expects the receiver to be able to deduce the decoder to shift to based on this. We refer to this setting as synchronization through "initial setup", analogous to the usage of "trusted initial setup" in distributed computing.

In typical emergent communication settings, the sender and receiver are assumed to not have any shared meaning of messages but instead synchronize on their meanings during their joint interaction. We provide a protocol that does not assume any such shared meaning of messages in Proposition 5.5. We refer to this setting as "plain mode" to contrast with "initial setup".

This protocol achieves a communication regret of $O(T^{1/2}MN^{3/2}(\log N)^{1/2})$. That is, the same optimal dependence on T but worse dependence on M, N compared to the protocol in initial setup given in Theorem 5.4.

Proposition 5.5. *(Synchronization in plain mode) Given any online algorithm $\mathcal{A}$ over the space $\Sigma \times \mathrm{P}$ of encoder-decoder pairs, there are sender and receiver algorithms that don't rely on initial shared meaning of messages that achieve communication regret at most*

$$R_T \leq R_T^{ext}(\mathcal{A}) + \mathcal{O}\left(MN^3 \log(1/\delta)\right) \cdot S_T(\mathcal{A}),$$

with probability at least $1 - \delta$ for $\delta > 0$, where $S_T(\mathcal{A})$ is the number of switches made by $\mathcal{A}$.

There are efficient sender and receiver algorithms that don't rely on initial shared meanings of messages that result in expected communication regret

$$\mathbb{E}[R_T] \in \mathcal{O}\left(T^{1/2}MN^{3/2}(\log N)^{1/2}\right).$$

The plain mode protocol obtaining the bounds in Proposition 5.5 and its analysis are provided in Appendix F.4. The sender and receiver algorithms in this protocol have aspects designed specifically for communication. For instance, the sender needs to intentionally cause a mistake to reset the receiver if the receiver accidentally learned the wrong decoding for a message.

However, the algorithms don't rely on an initial shared meaning of messages. If an adversary changed the ordering of messages via a random permutation, the algorithms still converge to efficient communication. This is not true of the algorithms designed in Theorem 5.4.

6 General Utilities

In this section, we study the more general setting in which the agents are collaborating not to recover the state, but to achieve an arbitrary task. At every iteration after receiving a message, the receiver now plays actions from a set $\mathcal{A}$ of size A, and the reward function $r : \mathcal{A} \times \Omega \to [0, U]$ is an arbitrary bounded function of actions and realized states.

We follow a similar algorithmic recipe to design sender-receiver protocols in this general utilities settings as we did in the previous setting where the reward was simply the reconstruction accuracy (i.e., the equality indicator). Recall that the general principle was to design an algorithm $\mathcal{A}_{\text{stable}}$ that jointly optimizes over the

sender's and receiver's action spaces to minimize external regret while also having a low number of switches. This algorithm guided the sender's strategy. We then combined this with a receiver that minimizes tracking regret, as in Theorem 5.3, or modified $\mathcal{A}_{\text{stable}}$ to include some rounds to signal and communicate switches and combined this with a more bespoke receiver algorithm as in Theorem 5.4.

We can continue to use the same receiver algorithms for the game with more general utilities. The only difference is that the action space of the receiver is mappings from messages to actions rather than messages to states. However, we will need to design a new $\mathcal{A}_{\text{stable}}$ that the sender can use since we used the structure of reconstruction utilities to derive computationally efficient regret minimizing algorithms over the space of encoder-decoder pairs.

In fact, we show in Theorem C.3 that under general utilities, unless RP = NP, we will no longer be able to achieve vanishing communication regret, even when jointly optimizing over sender-receiver actions. Instead, we aim to minimize an approximate version of communication regret defined below.

Definition 6.1. For $\alpha \in [0, 1]$, the α-approximate communication regret of a sequence $\chi = (\omega_t^*, \sigma_t, \rho_t)_{t=1}^T$, written $R_T^\alpha(\chi)$, is:

$$\alpha \cdot \max_{\sigma \in \Sigma, \rho \in \mathrm{P}} \sum_{t=1}^{T} \mathbb{1}\{\rho(\sigma(\omega_t^*)) = \omega_t^*\} - \sum_{t=1}^{T} \mathbb{1}\{\rho_t(\sigma_t(\omega_t^*)) = \omega_t^*\}.$$

This notion of a no-approximate regret algorithm has been studied before in various settings where finding the optimal solution is statistically tractable but computationally intractable, such as Kakade et al. [2007], Roughgarden and Wang [2018], Emamjomeh-Zadeh et al. [2021].

Theorem 6.2 demonstrates the existence of a pair of sender and receiver algorithms that achieve sublinear $(1 - 1/e)$-approximate communication regret. The key insight for the design of these algorithms is the submodular structure of the communication game with general utilities.

Theorem 6.2. *There are poly(A, N, T) sender and receiver algorithms for the communication game with general utilities that can achieve a $(1 - 1/e)$-approximate communication regret of*

$$\mathcal{O}\left(SM^2\sqrt{T \log N}\right),$$

where $S \in \mathbb{R}$ is such that for all $\omega \in \Omega$, the sum of utilities is bounded by S: $\sum_{a \in \mathcal{A}} r(a, \omega) \leq S$.

Proof sketch. We follow the same recipe to design the sender and receiver algorithms as we did in the previous settings. That is, we first find an algorithm that minimizes communication regret in the centralized communication setting where the sender and receiver strategies are jointly optimized. This algorithm also makes a small number of switches with high probability. The small number of switches allows us to design decentralized sender and receiver algorithms that get synchronized after each switch. We show this in more detail in Appendix C

In the rest of this proof sketch, we will describe an algorithm for the centralized communication game with general utilities. The algorithm relies on the submodular structure of rewards in the communication game with general utilities which we will establish below.

Submodular rewards in every round. We will show that the utility in every round is a monotone, submodular function.

Just as in the proof of Proposition 4.2, we can reduce the problem to optimizing over deterministic mappings from messages to actions that the sender would want to puppeteer the receiver to use. The sender's encoder is the best response to this mapping. We can again represent these deterministic mappings by their image sets. Therefore, subsets of the action space $\mathcal{A}$ of size $\leq k$ forms a representation of the action space of the centralized communication game.

The reward function at round t depends on the state ω_t^* realized at round t. Let us denote this function by $V_t : 2^{\mathcal{A}} \to \mathbb{R}$. $V_t(A)$ is the utility of the decoder with image set A and its best-response encoder. The best-response encoder is the one that encodes ω_t^* to the action in A yielding the highest reward r i.e., $\text{argmax}_{a \in A} r(a, \omega_t^*)$. Hence we can express V_t for any $A \subseteq \mathcal{A}$ with $|A| \leq k$ as:

$$V_t(A) = \max_{a \in A} r(a, \omega_t^*).$$

Finding the subset of actions resulting in the highest V_t is a special instance of the *segmentation problem* studied by Kleinberg et al. [2004], and the algorithm presented below can be seen as an instantiation of their greedy algorithm in section 4. V_t is clearly monotone since adding more actions to the image set can only increase utility. It is also the maximum value over a set and hence is submodular. By Buchbinder et al. [2014](Theorem 3.1), we can find a subset of $\mathcal{A}$ that is at least $(1 - 1/e)$ times the optimal value of V_t.

Algorithm 1 A No-Approximate-Regret Algorithm for the General Utility Centralized Communication Game (parameterized by ϵ, α).

1: Initialize M different copies of the Follow the Perturbed Leader algorithm (FTPL) that each pick actions in $\mathcal{A}$, each mini-batched into groups of α. Each copy picks an initial perturbation $p_i \sim \text{Unif}[0, 1/\epsilon]^M$.
2: Nature commits to a sequence of T states $\omega_1, \ldots, \omega_T$ before the game starts, hidden to the sender and receiver.
3: **for** $t \leftarrow 1$ to T **do**
4: Each ith instance of FTPL suggests action $a_i^{(t)}$.
5: The sender will play an encoding scheme that assigns to action $a_i^{(t)}$ a unique message.
6: The state ω_t is revealed to the sender, and the sender sends a message associated with an action in $\left\{a_1^{(t)}, \ldots, a_M^{(t)}\right\}$ that maximizes utility at state ω_t.
7: Let $V_t(A) = \max_{a \in A} r(a, \omega_t)$. The payoff for each action $a \in \mathcal{A}$ given to the ith copy of FTPL is

$$V_t\left(\left\{a_1^{(t)}, \ldots, a_i^{(t)}\right\}\right) - V_t\left(\left\{a_1^{(t)}, \ldots, a_{i-1}^{(t)}\right\}\right).$$

8: **end for**

Algorithm for centralized communication game. Since each round's utility function is monotone and submodular, using methods developed for online submodular optimization Streeter and Golovin [2008], we can achieve vanishing $(1 - 1/e)$-approximate regret.

The full details of the algorithm that achieves this is provided in Algorithm 1. Here we give a brief description. The algorithm maintains M copies of a no-regret algorithm, each having action set $\mathcal{A}$. Copy i is used to decide the target action for the receiver upon seeing message m_i. At round t when state ω_t^* is realized, the sender elicits the actions recommended by each of the M copies $A_t = (a_i^{(t)})_{i=1}^M$. The sender chooses the message $i \in [M]$ corresponding to the action $a_i^{(t)}$ with the highest reward for state ω_t^* i.e., the action $\text{argmax}_{i \in [M]} r(a_i^{(t)}, \omega_t^*)$.

The sender receives reward $V_t(A_t) = \max_{a \in A_t} r(a, \omega_t^*)$ and uses this to allocate rewards to each no-regret algorithm copy and update the copy using this allocated reward. The reward for copy i is $V_t(\{a_1^{(t)}, \ldots, a_i^{(t)}\}) - V_t(\{a_1^{(t)}, \ldots, a_{i-1}^{(t)}\})$.

Our algorithm flexibly takes any no-regret algorithm as a black box. By feeding in algorithms with a small number of switches [Kalai and Vempala, 2005], the resultant algorithm for online submodular optimization simultaneously has vanishing approximate regret and few switches. □

7 Discussion and Concluding Remarks

Our work initiates the study of algorithmic principles that drive the emergence of communication in multi-agent learning environments. We believe that our online communication game model provides a natural framework for exploring a range of algorithmic questions related to the role of training dynamics in emergent communication.

The properties of stability and adaptivity we identify in our work as sufficient conditions for communication to emerge can be incorporated into practical algorithms in a number of ways. For example, learning algorithms can be made more stable through regularization, decreased learning rates, and delayed updates. They can be made more adaptive by increased learning rates and restarting periodically. Indeed, previous empirical work [Chaabouni et al., 2022, Li and Bowling, 2019] demonstrates that incorporating some of these properties in emergent communication settings leads to improved communication. However the improvement has not been attributed to either of these properties. By explicitly and systematically identifying properties of good training algorithms, we hope to offer a more principled approach for the design of training algorithms.

There are a number of open questions and possible extensions of our framework. An open question is about the tightness of the communication regret rates. The algorithms we provide achieve a communication regret rate of $\Theta(T^{1/2})$ that is tight in T, the number of rounds. However, the tightness of dependence on the number of actions and messages is unknown, and so an open question is to derive lower bounds in terms of the number of states and messages.

There are many possible extensions to our framework to capture more complex communication settings. One extension is to games with continuous states and action spaces. Another extension is to games where both players have private information that they need to communicate.

8 Acknowledgments

The authors thank Brian Lee for helpful discussions and Cassidy Laidlaw for feedback on drafts. This work was supported by a grant from Open Philanthropy to the Center for Human-Compatible Artificial Intelligence at UC Berkeley.

References

Jason Altschuler and Kunal Talwar. Online learning over a finite action set with limited switching. In *Conference On Learning Theory*, pages 1569–1573. PMLR, 2018.

Raman Arora, Ofer Dekel, and Ambuj Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. *arXiv preprint arXiv:1206.6400*, 2012.

William Brown, Jon Schneider, and Kiran Vodrahalli. Is learning in games good for the learners? *Advances in Neural Information Processing Systems*, 36, 2024.

Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1433–1452. SIAM, 2014.

Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

Nicolo Cesa-Bianchi, Pierre Gaillard, Gábor Lugosi, and Gilles Stoltz. Mirror descent meets fixed share (and feels no regret). *Advances in Neural Information Processing Systems*, 25, 2012.

Nicolo Cesa-Bianchi, Ofer Dekel, and Ohad Shamir. Online learning with switching costs and other adaptive adversaries. *Advances in Neural Information Processing Systems*, 26, 2013.

Nicolo Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Regret minimization for reserve prices in second-price auctions. *IEEE Transactions on Information Theory*, 61(1):549–564, 2014.

Rahma Chaabouni, Florian Strub, Florent Altché, Eugene Tarassov, Corentin Tallec, Elnaz Davoodi, Kory Wallace Mathewson, Olivier Tieleman, Angeliki Lazaridou, and Bilal Piot. Emergent communication at scale. In *International conference on learning representations*, 2022.

Francis Chu and Joseph Halpern. On the np-completeness of finding an optimal strategy in games with common payoffs. *International Journal of Game Theory*, 30:99–106, 2001.

Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R McKee, Joel Z Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative ai. *arXiv preprint arXiv:2012.08630*, 2020.

Amit Daniely, Alon Gonen, and Shai Shalev-Shwartz. Strongly adaptive online learning. In *International Conference on Machine Learning*, pages 1405–1411. PMLR, 2015.

Ehsan Emamjomeh-Zadeh, Chen-Yu Wei, Haipeng Luo, and David Kempe. Adversarial online learning with changing action sets: Efficient algorithms with approximate regret bounds. In *Algorithmic Learning Theory*, pages 599–618. PMLR, 2021.

Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

Jakob N Foerster, Yannis M Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate to solve riddles with deep distributed recurrent q-networks. *arXiv preprint arXiv:1602.02672*, 2016.

Michael Franke. Interpretation of optimal signals. *New perspectives on games and interaction*, pages 297–310, 2009a.

Michael Franke. *Signal to act: Game theory in pragmatics*. University of Amsterdam, 2009b.

Itzhak Gilboa and Eitan Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1(1):80–93, 1989.

Laura Graesser, Kyunghyun Cho, and Douwe Kiela. Emergent linguistic phenomena in multi-agent communication games. *arXiv preprint arXiv:1901.08706*, 2019.

Shangmin Guo, Yi Ren, Serhii Havrylov, Stella Frank, Ivan Titov, and Kenny Smith. The emergence of compositional languages for numeric concepts through iterated learning in neural agents. *arXiv preprint arXiv:1910.05291*, 2019.

Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *Advances in neural information processing systems*, 30, 2017.

Mark Herbster and Manfred K Warmuth. Tracking the best expert. *Machine learning*, 32(2):151–178, 1998.

Athul Paul Jacob, Gabriele Farina, and Jacob Andreas. Regularized conventions: Equilibrium computation as a model of pragmatic reasoning. *arXiv preprint arXiv:2311.09712*, 2023.

Gerhard Jäger. Game dynamics connects semantics and pragmatics. In *Game theory and linguistic meaning*, pages 103–117. Brill, 2007.

Gerhard Jäger. Game theory in semantics and pragmatics. *Semantics: An international handbook of natural language meaning*, 3:2487–2516, 2012.

Sham M Kakade, Adam Tauman Kalai, and Katrina Ligett. Playing games with approximation algorithms. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 546–555, 2007.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Michael Kapralov, Ian Post, and Jan Vondrák. Online submodular welfare maximization: Greedy is optimal. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 1216–1225. SIAM, 2013.

Jooyeon Kim and Alice Oh. Emergent communication under varying sizes and connectivities. *Advances in Neural Information Processing Systems*, 34:17579–17591, 2021.

Simon Kirby. Natural language from artificial life. *Artificial life*, 8(2):185–215, 2002.

Simon Kirby, Tom Griffiths, and Kenny Smith. Iterated learning and the evolution of language. *Current opinion in neurobiology*, 28:108–114, 2014.

Jon Kleinberg, Christos Papadimitriou, and Prabhakar Raghavan. Segmentation problems. *J. ACM*, 51(2): 263–280, March 2004. ISSN 0004-5411. doi: 10.1145/972639.972644. URL https://doi.org/10.1145/972639.972644.

Angeliki Lazaridou and Marco Baroni. Emergent multi-agent communication in the deep learning era. *arXiv preprint arXiv:2006.02419*, 2020.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. *arXiv preprint arXiv:1612.07182*, 2016.

Joshua Letchford, Vincent Conitzer, and Kamesh Munagala. Learning and approximating the optimal strategy to commit to. In *Algorithmic Game Theory: Second International Symposium, SAGT 2009, Paphos, Cyprus, October 18-20, 2009. Proceedings 2*, pages 250–262. Springer, 2009.

David Lewis. *Convention: A philosophical study*. John Wiley & Sons, 2008.

Fushan Li and Michael Bowling. Ease-of-teaching and language structure from emergent communication. *Advances in neural information processing systems*, 32, 2019.

Ryan Lowe, Jakob Foerster, Y-Lan Boureau, Joelle Pineau, and Yann Dauphin. On the pitfalls of measuring emergent communication. *arXiv preprint arXiv:1903.05168*, 2019.

Haipeng Luo and Robert E Schapire. Achieving all with no parameters: Adanormalhedge. In *Conference on Learning Theory*, pages 1286–1304. PMLR, 2015.

Binghui Peng, Weiran Shen, Pingzhong Tang, and Song Zuo. Learning optimal strategies to commit to. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2149–2156, 2019.

Limor Raviv, Antje Meyer, and Shiri Lev-Ari. Compositional structure can emerge without generational transmission. *Cognition*, 182:151–164, 2019a.

Limor Raviv, Antje Meyer, and Shiri Lev-Ari. Larger communities create more systematic languages. *Proceedings of the Royal Society B*, 286(1907):20191262, 2019b.

Yi Ren, Shangmin Guo, Serhii Havrylov, Shay Cohen, and Simon Kirby. Enhance the compositionality of emergent language by iterated learning. In *3rd NeurIPS Workshop on Emergent Communication (EmeCom@ NeurIPS 2019). URL https://papers. nips. cc/book/advances-in-neural-information-processing-systems-32-2019*, 2019.

Mathieu Rita, Rahma Chaabouni, and Emmanuel Dupoux. " lazimpa": Lazy and impatient neural agents learn to communicate efficiently. *arXiv preprint arXiv:2010.01878*, 2020.

Mathieu Rita, Corentin Tallec, Paul Michel, Jean-Bastien Grill, Olivier Pietquin, Emmanuel Dupoux, and Florian Strub. Emergent communication: Generalization and overfitting in lewis games. *Advances in neural information processing systems*, 35:1389–1404, 2022.

Tim Roughgarden and Joshua R Wang. An optimal learning algorithm for online unconstrained submodular maximization. In *Conference On Learning Theory*, pages 1307–1325. PMLR, 2018.

Rohin Shah, Pedro Freire, Neel Alex, Rachel Freedman, Dmitrii Krasheninnikov, Lawrence Chan, Michael D Dennis, Pieter Abbeel, Anca Dragan, and Stuart Russell. Benefits of assistance over reward learning, 2020.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

John Maynard Smith and David Harper. *Animal signals*. Oxford University Press, 2003.

Luc Steels. The synthetic modeling of language origins. *Evolution of communication*, 1(1):1–34, 1997.

Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. *Advances in Neural Information Processing Systems*, 21, 2008.

Peter E Trapa and Martin A Nowak. Nash equilibria for an evolutionary language game. *Journal of mathematical biology*, 41(2):172–188, 2000.

Rotem Ben Zion, Boaz Carmeli, Orr Paradise, and Yonatan Belinkov. Semantics and spatiality of emergent communication. *arXiv preprint arXiv:2411.10173*, 2024.

Tijana Zrnic, Eric Mazumdar, Shankar Sastry, and Michael Jordan. Who leads and who follows in strategic classification? *Advances in Neural Information Processing Systems*, 34:15257–15269, 2021.

Appendix

Table of Contents

A Adaptive Adversaries Are Too Powerful

A key assumption in this work is that the environment cannot adaptively pick states based on the choices of the sender and receiver. The reason for this is that the adversary becomes too powerful otherwise, removing any hope the learners have of achieving sub-linear regret. This is analogous to results in the prediction from experts with switching costs literature, where the learner must suffer linear regret with an adaptive adversary [Altschuler and Talwar, 2018].

The adversary's strategy will be to exploit the fact that the Sender and Receiver play moves that are uncorrelated given the history of the game, which the adversary also has access to. Adaptive adversaries can exploit uncoordinated agents by taking advantage of the fact that it is impossible to randomize over best response pairs in a way that beats an adaptive adversary when actions are independently chosen.

First, we prove a lower bound when the sender is one message short from perfect communication.

Lemma A.1. *In the communication game with an adaptive adversary, when $N \geq 3$ and $M = N - 1$, any learning algorithm must achieve a regret of at least $\frac{N-2}{2N} \cdot T$.*

Proof. First we will lower bound optimal utility. Notice that regardless of the states played by the adversary, in hindsight, there must always exist an encoder-decoder pair that achieves $\frac{N-1}{N}T$ regret. Indeed, let $\omega_1, \ldots, \omega_T$ be the sequence of states played by the adversary. There must exist a state ω that appears at most $\frac{T}{N}$ times. The rest of the states must appear at least $\frac{N-1}{N} \cdot T$ times. Let σ be the encoder that assigns to each of these states a unique one of the $N-1$ messages, and ρ be the decoder that reverses this. This correctly recovers the state when ω doesn't appear, so achieves a utility of at least $\frac{N-1}{N} \cdot T$.

Now we will lower bound the utility the agents can achieve during the learning process. On step t over the T steps, let $\mathcal{H}_t = (\omega_1, \sigma_1, \rho_1, \ldots, \omega_{t-1}, \sigma_{t-1}, \rho_{t-1})$ be the history of states, encoders, and decoders played. The key property that we will exploit is that the moves the sender and receiver play are independent, so that $\sigma_t \perp \rho_t \mid \mathcal{H}_t$.

Given a σ_{t-1} and ρ_{t-1}, what state should the adversary play? The expected utility that the sender and receiver achieve, when the adversary plays state $\omega \in \Omega$ is the probability that the agents correctly recover ω:

$$\begin{aligned}
\mathbb{P}_{\sigma_t, \rho_t}\left(\omega = \rho_t(\sigma_t(\omega)) \mid \mathcal{H}_t\right) &= \sum_{m \in \mathcal{M}} \mathbb{P}_{\sigma_t, \rho_t}\left(\omega = \rho_t(m) \text{ and } m = \sigma_t(\omega) \mid \mathcal{H}_t\right) \\
&= \sum_{m \in \mathcal{M}} \mathbb{P}_{\rho_t}\left(\omega = \rho_t(m) \mid \mathcal{H}_t\right) \cdot \mathbb{P}_{\sigma_t}\left(m = \sigma_t(\omega) \mid \mathcal{H}_t\right) \\
&= \mathop{\mathbb{E}}_{m \sim \sigma_t(\omega) \mid \mathcal{H}_t}\left[\mathbb{P}_{\rho_t}\left(\omega = \rho_t(m) \mid \mathcal{H}_t\right)\right].
\end{aligned}$$

We will show that there must always exist a state $\omega \in \Omega$ such that

$$\mathop{\mathbb{E}}_{m \sim \sigma_t(\omega) \mid \mathcal{H}_t}\left[\mathbb{P}_{\rho_t}\left(\omega = \rho_t(m) \mid \mathcal{H}_t\right)\right] \leq \frac{1}{2}.$$

Order the states $\omega_1, \ldots, \omega_N$. If this is true for any of the first $N-1$, we are done. Otherwise, it is the case that for all i from 1 to $N-1$,

$$\mathop{\mathbb{E}}_{m \sim \sigma_t(\omega_i) \mid \mathcal{H}_t}\left[\mathbb{P}_{\rho_t}\left(\omega_i = \rho_t(m) \mid \mathcal{H}_t\right)\right] > \frac{1}{2}.$$

Therefore, for each i from 1 to $N-1$, there must exist some $m_i \in \mathcal{M}$ such that

$$\mathbb{P}_{\rho_t}\left(\omega_i = \rho_t(m_i) \mid \mathcal{H}_t\right) > \frac{1}{2}. \tag{1}$$

For any i from 1 to n, for any state $\omega \in \Omega$ with $\omega \neq \omega_i$:

$$\begin{aligned}\mathbb{P}_{\rho_t}\left(\omega = \rho_t(m_i) \mid \mathcal{H}_t\right) &= 1 - \mathbb{P}_{\rho_t}\left(\omega \neq \rho_t(m_i)\right) \\ &\leq 1 - \mathbb{P}_{\rho_t}\left(\omega_i = \rho_t(m_i)\right) \\ &< 1 - \frac{1}{2} = \frac{1}{2}.\end{aligned}$$

So, for any state that is not ω_i,

$$\mathbb{P}_{\rho_t}\left(\omega = \rho_t(m_i) \mid \mathcal{H}_t\right) < \frac{1}{2}. \tag{2}$$

By Equations (1) and (2), no two m_i can be equal. Since this is true for $N - 1 = M$ states, and every state is assigned a unique message by the above, every message appears in the set of $m_1, \ldots, m_{N-1}$. That is, there is a perfect matching between the set of all messages and the first $N - 1$ states, where when a message m_i is played, it must correspond to some state ω_i with high probability. Therefore, for the final state ω_N, every message is unlikely to correspond to it, i.e. for every message $m \in M$,

$$\mathbb{P}_{\rho_t}\left(\omega_N = \rho_t(m_i) \mid \mathcal{H}_t\right) \leq \frac{1}{2},$$

and so

$$\mathop{\mathbb{E}}_{m \sim \sigma_t(\omega_N) \mid \mathcal{H}_t}\left[\mathbb{P}_{\rho_t}\left(\omega_N = \rho_t(m_i) \mid \mathcal{H}_t\right)\right] \leq \frac{1}{2}.$$

So, the adversary can always play a state ω that forces the expected utility the agents achieve to be at most $\frac{1}{2}$. Whereas in hindsight, the agents could have achieved an average of $\frac{N-1}{N}$ per step, resulting in an expected regret of at least:

$$\mathbb{E}\left[\max_{\sigma,\rho} \sum_{t=1}^{T} \mathbb{1}(\omega_t = \rho(\sigma(\omega_t)) - \sum_{t=1}^{T} \mathbb{1}(\omega_t = \rho_t(\sigma_t(\omega_t))\right] \geq \frac{N-1}{N} \cdot T - \frac{1}{2} \cdot T = \frac{N-2}{2N} \cdot T. \qquad \square$$

To extend this result to the case where the number of messages is arbitrary, the adversary can beforehand commit to only using a subset of the states. It is interesting that even if the learners know which subset of the states the adversary has committed to using, they still cannot achieve sublinear regret.

Theorem A.2. *In the communication game with an adaptive adversary, when $N \geq 3$ and $1 < M < N$, any learning algorithm must suffer a regret of at least $\frac{M-1}{2M+2}T$.*

Proof. When M may be arbitrary, the adversary may just commit upfront to only sending $M + 1 \leq N$ states, fixed at the start arbitrarily. Since $M \geq 2$, this means that there are $M + 1 \geq 3$ states, so the lower bound in Lemma A.1 directly applies. $\square$

For the learning algorithms we derive to be nontrivial, we then must assume that the environment is oblivious to the actions of the sender and receiver.

B Better Bounds for Stochastic Environments

In the stochastic setting where the distribution over states is the same in all rounds, nature picks a distribution over states and sticks to it throughout the game, we can prove that a regret of $\tilde{\mathcal{O}}\left(T^{1/2}\right)$ is achievable.

The improved regret bounds are made possible through improved switching regret in the stochastic setting compared to the adversarial setting.

This translates to an improved communication regret across all settings—adaptive receiver, synchronization with an initial setup, and synchronization in plain mode by plugging in the improved switching regret bounds into Theorems 5.3, 5.4, and Proposition 5.5.

Proposition B.1. *There is an efficient algorithm that allows the sender and receiver to learn a language with regret* $\mathbb{E}[R_T] \in \tilde{\mathcal{O}}\left(MT^{1/2}(\log N)^{1/2}\right)$ *in the centralized communication game that switches at most* $1 + \log\log T$ *times.*

Proof sketch. We present a method that achieves a regret of $T^{1/2}$ that only requires the sender to switch their strategy $\mathcal{O}(\log\log T)$ times. This is inspired by Cesa-Bianchi et al. [2013, 2014].

The key idea is to have a sequence of stages where stage s lasts for T_s steps, starting with $s = 0$:

1. Use the empirical estimate of the probability of each state so far to estimate the optimal communication strategy.
2. Commit to this for T_s steps, all the while gathering more data on the number of times each state appears.
3. After T_s steps have run, increment s and go back to step 1 if we haven't run for at least T steps yet.

Set the length of each stage to be $T_s = T^{1-2^{-s}}$. In Appendix F.5, we show that this is just the right amount of stability to get $\tilde{\mathcal{O}}\left(T^{1/2}\right)$ regret. □

Using Proposition B.1 as a backbone, we can derive regret bounds for each of the various settings we consider.

Corollary B.2. *With a stochastic environment, a stable sender and an adaptive receiver (Definition 5.2) can achieve regret of* $\mathcal{O}\left(T^{1/2}N^{1/2}M\sqrt{\log\log T}\right)$. *There exists an efficient algorithm for the sender and receiver that achieves a regret of* $\tilde{\mathcal{O}}\left(T^{1/2}M(\log N)^{1/2} + M\log N\right)$. *Finally, the sender and receiver can coordinate without initial shared meanings of messages efficiently and achieve a regret of* $\tilde{\mathcal{O}}\left((T^{1/2} + N^3)M\right)$.

The three parts of the Corollary directly follow by applying Proposition B.1 to Theorem 5.3, Theorem 5.4, and Proposition 5.5 respectively. We can apply Proposition B.1 to Theorem 5.3 because the number of switches is deterministically $\log\log T + 1$.

Switching induces a cost, so a natural question that arises is whether we can get away with less than $\mathcal{O}(\log\log T)$ switches and still achieve sub-linear regret. We answer this in the affirmative in Appendix D.2, introducing a protocol that achieves $\tilde{\mathcal{O}}\left(MT^{2/3}\right)$ regret with only a single switch.

For general utility communication games, with similar techniques, we can get a linear dependence on M, and an improved dependence on the utility, with an upper bound linear in the infinity norm of the reward rather than the ℓ_1 norm.

Theorem B.3. *There is an efficient no-regret algorithm that can achieve a* $(1 - 1/e)$*-approximation of optimal communication with an arbitrary utility* $r : \Omega \times \mathcal{A} \to [0, U]$ *in stochastic environments with regret* $\tilde{\mathcal{O}}\left(UMT^{1/2}\right)$.

C General Utilities - Extended Results

In this section, we study the more general setting where the agents are collaborating not to recover the state but to achieve arbitrary tasks. At every iteration after receiving a message, the receiver now plays actions from a set $\mathcal{A}$, and the reward function $r : \Omega \times \mathcal{A} \to [0, U]$ is arbitrary.

The insights we gained solving the problem in the case where the reward was simply the reconstruction accuracy (i.e., the equality indicator) can be used to create an algorithm that works for general reward functions. The solutions it arrives at can be suboptimal, however. We then show that this is in fact the best you can hope for, proving that any no-regret learning algorithm that achieves better utility must be computationally intractable.

The two important parameters that will control the difficulty of this problem are the number of actions $A := |\mathcal{A}|$ and the max sum of utilities in each state $S := \max_{\omega\in\Omega} \sum_{a\in\mathcal{A}} r(\omega, a)$.

C.1 Upper Bounds Via a Reduction to the Equality Game

We can use the insights gained in the sections above to design no-regret algorithms that efficiently achieve a $(1 - 1/e)$-approximation of optimal play.

Definition 6.1. For $\alpha \in [0, 1]$, the α-approximate communication regret of a sequence $\chi = (\omega_t^*, \sigma_t, \rho_t)_{t=1}^T$, written $R_T^\alpha(\chi)$, is:

$$\alpha \cdot \max_{\sigma \in \Sigma, \rho \in \mathrm{P}} \sum_{t=1}^{T} \mathbb{1}\{\rho(\sigma(\omega_t^*)) = \omega_t^*\} - \sum_{t=1}^{T} \mathbb{1}\{\rho_t(\sigma_t(\omega_t^*)) = \omega_t^*\}.$$

This notion of a no-approximate regret algorithm has been studied before in various settings where finding the optimal solution is statistically tractable but computationally intractable, such as Kakade et al. [2007], Roughgarden and Wang [2018], Emamjomeh-Zadeh et al. [2021].

The key idea of the general reduction is to associate each message with an action, and have the sender puppeteer the receiver. To find the right actions to assign to each message, we write the problem as one of maximizing a monotone, submodular set function. We then use the insights of the previous sections to find the correct times for the sender to update the encoder with its best guess of what maximizes this submodular function.

When the states come not from a fixed, hidden distribution, but from an adversary that is oblivious to the actions of the sender and receiver, we can design similar algorithms by combining the insights above with those of Streeter and Golovin [2008] and Kalai and Vempala [2005].

Theorem 6.2. *There are poly(A, N, T) sender and receiver algorithms for the communication game with general utilities that can achieve a $(1 - 1/e)$-approximate communication regret of*

$$\mathcal{O}\left(SM^2\sqrt{T \log N}\right),$$

where $S \in \mathbb{R}$ is such that for all $\omega \in \Omega$, the sum of utilities is bounded by S: $\sum_{a \in \mathcal{A}} r(a, \omega) \leq S$.

Proof. Without the loss of generality, we can assume $S = 1$. Because it is bounded, we can scale the reward function until this holds by dividing all utilities by S. In the worst case, regret is S times more than the normalized game.

The key realization needed to solve this game is that it is a submodular maximization problem in disguise. Recall that there always exists an optimal deterministic policy. A deterministic decoder will always map the M messages to M fixed actions. So, the task of finding the optimal communication scheme boils down to finding these M actions we would like the receiver to play. Once we have these, we can have the sender at every turn tell the receiver which of these M actions maximizes utility on their current observed state.

To solve for this, define a function $V : 2^{\mathcal{A}} \to \mathbb{R}$, that when given a collection of actions finds the expected value of assigning each of them a unique message:

$$V(\{a_1, \ldots, a_k\}) = \mathop{\mathbb{E}}_{\omega \sim \mathcal{D}} \left[\max_i r(a_i, \omega)\right].$$

This is a monotone function as increasing the number of messages we send can only improve reward. Moreover, for any sets of actions $A, B \subseteq \mathcal{A}$, we can write

$$V(A) + V(B) = \mathop{\mathbb{E}}_{\omega \sim \mathcal{D}} \left[\max_{a \in A} r(a, \omega)\right] + \mathop{\mathbb{E}}_{\omega \sim \mathcal{D}} \left[\max_{a' \in B} r(a', \omega)\right].$$

For any state $\omega \in \Omega$, note that $\max_{a \in A \cup B} r(a, \omega)$ must equal at least one of $\max_{a \in A} r(a, \omega)$ and $\max_{a \in B} r(a, \omega)$. Moreover, since $A \cap B$ is a subset of both A and B, it must be that $\max_{a \in A \cap B} r(a, \omega)$ is at most $\max_{a \in A} r(a, \omega)$

and at most $\max_{a \in B} r(a, \omega)$. Therefore,

$$\begin{aligned} V(A) + V(B) &= \mathop{\mathbb{E}}_{\omega \sim \mathcal{D}} \left[\max_{a \in A} r(a, \omega) + \max_{a' \in B} r(a', \omega) \right] \\ &\geq \mathop{\mathbb{E}}_{\omega \sim \mathcal{D}} \left[\max_{a \in A \cap B} r(a, \omega) + \max_{a' \in A \cup B} r(a', \omega) \right] \\ &= V(A \cap B) + V(A \cup B), \end{aligned}$$

proving submodularity. As a remark, to solve the offline problem, we would need to maximize V under the constraint that the input is of size at most M. By Buchbinder et al. [2014, Theorem 3.1], there exists an algorithm that can compute a $(1 - 1/e)$-approximation of the optimal set efficiently.

Now all we need is an online algorithm for maximizing submodular functions with a small number of switches. Streeter and Golovin [2008] propose a general framework that we utilize to design algorithms for online monotone submodular maximization with switching costs and cardinality constraints.

To obtain an algorithm that works in the adversarial setting, we won't care about maximizing $V(\{a_1, \ldots, a_k\}) = \mathbb{E}_{\omega \sim \mathcal{D}} [\max_i r(a_i, \omega)]$ as above, instead we'll design an algorithm to maximize $\sum_{t=1}^{T} V_t(\{a_1, \ldots, a_k\})$, where each $V_t(\{a_1, \ldots, a_k\}) = \max_i r(a_i, \omega_t)$. The adversary chooses the sequence of ω_t before the game begins.

The algorithm will rely on the Follow the Perturbed Leader (FTPL) algorithm in Kalai and Vempala [2005], mini-batched into groups of α as is done in Corollary F.2. Using this, we can develop Algorithm 1, a no-approximate-regret online monotone submodular function maximization algorithm with switching costs, cardinality constraints, and an oblivious adversary, with optimal regret rates in T.

Notice that the payoffs of the ith instance of FTPL depends on the choices made by nature and the first $i - 1$ instances of FTPL, but not on their own history of choices. So, from the perspective of each FTPL instance, they are playing against oblivious adversaries.

This simulates the greedy algorithm for maximizing monotone submodular functions under cardinality constraints in an online manner. We choose copies FTPL to decide actions, for the purpose of being able to bound the number of times the sender will change their encoding scheme. This isn't strictly necessary however, and any online learning algorithm that achieves low regret with oblivious adversaries can be used here.

By Corollary F.2, each instance of FTPL achieves a regret of $2\alpha\sqrt{2MT}$ with $1/\alpha\sqrt{2MT}$ switches.

By Streeter and Golovin [2008, Lemma 3], we can bound the $(1 - 1/e)$-expected-regret of the sender in the centralized, general-utility game by the sum of the regrets of each instance, meaning Algorithm 1 achieves a regret of $2M\alpha\sqrt{2MT}$ in expectation. We switch whenever any one of the M experts switch, so we switch at most $M/\alpha\sqrt{2MT}$ times. The cost to each switch can be taken down to, by Theorem 5.4, $M \log_M(N)$. Using the coordination protocol in Theorem 5.4, this means that we incur at most a

$$\frac{M^2 \log(N)}{\alpha} \sqrt{2TM}$$

cost from switching. Set $\alpha = \sqrt{M \log N}$, and the expected regret that Algorithm 1 achieves is

$$3M^2 \sqrt{T \log(N)}.$$

□

Corollary C.1. *Suppose the sender and receiver follow the stable sender, adaptive receiver protocol in a communication game with general utilities (Definition 5.2) using the algorithms $\mathcal{A}_{\text{stable}}, \mathcal{A}_{\text{adaptive}}$, then the communication regret is at most*

$$R_T \leq R_T^{ext}(\mathcal{A}_{\text{stable}}) + R_T^{track}(\mathcal{A}_{\text{adaptive}}; S_T(\mathcal{A}_{\text{stable}}; \delta)) + \delta T,$$

for every $\delta > 0$, where $S_T(\mathcal{A}_{\text{stable}}; \delta)$ is the number of switches made by $\mathcal{A}_{\text{stable}}$ with probability at least $1 - \delta$ and R_T^{ext} and R_T^{track} denote external and tracking regrets.

There exist efficient algorithms $\mathcal{A}_{\text{stable}}, \mathcal{A}_{\text{adaptive}}$ that result in expected $(1 - 1/e)$-communication regret

$$\mathbb{E}[R_T] \in \mathcal{O}\left(ST^{2/3} M^{4/3} N^{1/3} \log(N)^{1/3} \log(MT)^{1/6} + SM^2 \log(N) \right).$$

C.2 Hardness Results for Arbitrary Utilities

The algorithm given above does in fact meet the theoretically optimal approximation factor.

When there are multiple best actions in each state, it may not be clear what the most efficient way to group states together is. This source of complexity on its own is enough to make the problem NP-hard. And not only is this problem hard, but the simpler problem with rewards restricted to be 0 or 1 is computationally intractable in the worst case. Because we can efficiently derandomize, allowing the outputted solutions to be stochastic doesn't make the problem easier.

Lemma C.2. *Computing any stochastic communication strategy that is an α-approximation of the optimal strategy, where $\alpha > 1 - 1/e$, in a communication game with rewards restricted to be 0 or 1 is NP-hard.*

With Lemma C.2 as the backbone, we can prove that efficient learning algorithms cannot do better than our Algorithm 1 under standard complexity-theoretic assumptions, even in the simplest setting where nature commits to a fixed distribution over observations beforehand.

Theorem C.3. *Unless RP = NP, for any $\alpha > 1-1/e$, any algorithm that runs in time poly(N, M) per iteration has α-approximate communication regret such that either $R_T^\alpha \in \omega(T^{1-\epsilon})$ for all $\epsilon > 0$ or $R_T^\alpha \notin$ poly(N, M).*

Kapralov et al. [2013] show a similar result, under the same complexity-theoretic assumption on the limits of randomness, for the problem of online welfare maximization.

Remark C.4. Theorem C.3 doesn't rule out the existence of very slow no-regret algorithms, for example, there could still exist a regret $\mathcal{O}\left(\frac{T}{\log T}\right)$ algorithm for the sender and receiver that achieves an approximation factor of $\alpha > 1 - 1/e$.

D Minimizing the Number of Switches

All else being equal, stable algorithms will outperform unstable algorithms. The natural follow up question is: what are the *most* stable learning algorithms? The goal of this section is to introduce an analogue of the Explore-then-Commit algorithm and prove guarantees on its performance.

D.1 A Fine-Grained Analysis of Switching Cost

The sender and receiver can coordinate well not only when switches are infrequent, but also when changes are local. Let $\mathsf{Ham}(\delta, \delta') = \sum_{m \in \mathcal{M}} \mathbb{1}(\delta(m) = \delta'(m))$ be the Hamming distance of two decoding schemes. We can give better bounds on switching cost in terms of the Hamming distance of the best responses of the sender's strategy $\mathsf{Ham}(\mathsf{BR}(\sigma_t), \mathsf{BR}(\sigma_{t+1}))$.

Proposition D.1. *The cost of switching from encoding scheme σ to encoding scheme σ' is at most $1 + \mathsf{Ham}(\mathsf{BR}(\sigma), \mathsf{BR}(\sigma'))(1 + \log_M(N))$.*

Proof. When switches are small, the sender and receiver may coordinate via a more intricate messaging scheme. When it is time for a predetermined switch, the sender sends a message detailing how many of the at most M outputs of the best response has changed in $\log_M(M) = 1$ message. For each message changed, the sender starts off by sending that message, then sending the optimal state to respond with in $\log_M(N)$ messages. This all in all incurs a switching cost of $1 + \mathsf{Ham}(\mathsf{BR}(\sigma_t), \mathsf{BR}(\sigma_{t+1}))(1 + \log_M(N))$. □

D.2 Capture then Commit

We can define the Capture-then-commit policy as follows:

1. For the first T' steps, maintain a counter for the number of times each state is received. During this phase, send the receiver arbitrary messages.

2. For the remaining $T - T'$ steps, play the current estimate of the optimal strategy. The counters for the number of times each state is observed are not updated anymore.

Using this we can get what seems to be optimal regret.

Proposition D.2. *Capture then Commit achieves a regret of* $\tilde{\mathcal{O}}\left(MT^{2/3}\right)$ *in the centralized communication game.*

Proof. This is an application of Lemma F.4. Aggregating the frequency of the first $T^{2/3}$ iterations gives us an encoder-decoder scheme in the centralized game with regret at most

$$\frac{\sqrt{2}\left(1 + M\sqrt{\log(NT^{1/3})}\right)}{T^{1/3}}$$

per step.

During the capture phase, we accumulate a regret of at most $T^{2/3}$. During the rest, we accumulate a regret of at most $\sqrt{2}T^{2/3} \cdot \left(1 + M\sqrt{\log(NT^{2/3})}\right)$, meaning in total $\tilde{\mathcal{O}}\left(T^{2/3}M\right)$. □

We can use the same techniques as before to generalize this to the decentralized communication game.

Theorem D.3. *The sender and receiver can play a joint policy that achieves a regret of* $\mathcal{O}\left(T^{2/3}M\sqrt{\log(NT)} + M\log(N)\right)$.

Proof. The capture then commit protocol only switches once, so the cost of switching is constant in T. Using the regret bound in the centralized game derived in Proposition D.2, and the switching cost derived in Theorem 5.4, we get a regret of $\mathcal{O}\left(T^{2/3}M\sqrt{\log(NT)} + M\log(N)\right)$. □

E Useful Results and Algorithms From Previous Online Learning Frameworks

E.1 Minmax lower bound

The tools used to show a minmax lower bound of $\Omega(\sqrt{T})$ regret in the standard online learning regret minimization setting can be used to derive a similar minmax lower bound of $\Omega(\sqrt{T})$ for communication regret. That is, we can show that for any pair of sender and receiver algorithms, there is an instance of the communication game that results in communication regret at least $\Omega(\sqrt{T})$. The lower bound is constructed through showing limitations on distinguishing between two Bernoulli distributions with means $1/2 + \varepsilon$ and $1/2 - \varepsilon$.

At a high level, we can view the general online learning problem as a special case of the online communication problem with just a single message that the sender can send. The receiver's problem in this case resembles the standard external regret minimization problem.

Proposition E.1 (Communication regret lower bound)**.** *For every sender-receiver protocol, there is an instance of the online communication game for which communication regret is* $\Omega(\sqrt{T})$.

Proof. We will show a reduction from the problem of distinguishing between two Bernoulli distributions to a the communication game.

The ε-Bernoulli distinction problem is the problem of given a distribution that is one of $\text{Bern}(1/2 - \varepsilon)$ or $\text{Bern}(1/2 + \varepsilon)$, determining which distribution it is.

Standard lower bounds for this problem state that the probability of success of any algorithm with T samples in the ε-Bernoulli distinction problem, is at most $1 - \exp(-\varepsilon^2 T/2)$.

Now we will show that these lower bounds imply a lower bound for communication regret by establishing an algorithmic reduction

Algorithm for distinction from algorithm for the communication game. Consider any sender-receiver algorithms for the communication game. We can construct an algorithm for the ε-Bernoulli distinction problem in the following way.

We can set up a communication game with a single message where the state is drawn according to the distribution D which is one of $\text{Bern}(1/2-\varepsilon)$ or $\text{Bern}(1/2+\varepsilon)$.

If the receiver predicts the state to be 1 at least $1/2$, we conclude that the distribution is $\text{Bern}(1/2+\varepsilon)$ and otherwise, we conclude the probability is $\text{Bern}(1/2+\varepsilon)$.

Let us analyze the probability of success of this approach and relate it to the communication regret.

Let the communication games induced by state generating distributions $\text{Bern}(1/2-\varepsilon)$, $\text{Bern}(1/2+\varepsilon)$ be G_ε, G'_ε respectively. For a sender-receiver protocol, let $C_1(T)$ be a random variable denoting the number of times the receiver predicts the state to be 1 in T rounds. For any sender-receiver pair π, we can write the communication regret under the games $G_\varepsilon, G'_\varepsilon$ as $R_T(\pi, G_\varepsilon)$ and $R_T(\pi, G'_\varepsilon)$ respectively.

We can bound both regrets in terms of the number of times the receiver predicts state 1 in the following way. $R_T(\pi, G_\varepsilon) \geq \mathbb{P}_{D_\varepsilon}(C_1(T) \leq T/2) \cdot T\varepsilon/2$ and $R_T(\pi, G'_\varepsilon) \geq \mathbb{P}_{D'_\varepsilon}(C_1(T) > T/2) \cdot T\varepsilon/2$.

$$\begin{aligned} R_T(\pi, G_\varepsilon) + R_T(\pi, G'_\varepsilon) &\geq \mathbb{P}_{D_\varepsilon}(C_1(T) \leq T/2)\frac{T\varepsilon}{2} + \mathbb{P}_{D'_\varepsilon}(C_1(T) > T/2)\frac{T\varepsilon}{2} \\ &\geq \frac{T\varepsilon}{2} \cdot \text{Probability of success in distinction problem} \\ &\geq \frac{T\varepsilon}{2}\exp(-\varepsilon^2 T/2). \end{aligned}$$

Choosing $\varepsilon = \max(1/2, \sqrt{1/4T})$ results in the lower bound of the proposition.

□

E.2 Bandits with Switching Cost

In the bandits with switching costs problem, in addition to the reward achieved from the action selected, there is a constant cost λ incurred every time the action selected in a round differs from the action selected in the previous round.

Any algorithm that minimizes external regret can be transformed to minimize switching regret which is external regret plus λ times the number of switches.

The transformation divides T rounds into batches each of length τ and queries the external regret minimizing at the start of the batch and selects the output as the action throughout the batch. At the end of the batch, the external regret minimizing algorithm is updated with the average reward obtained during the batch.

Theorem E.2 (Restatement of theorem from Cesa-Bianchi et al. [2013])**.** *Given an algorithm that achieves a regret of at most $R(T,k)$ dependence on the number of rounds T and number of actions k, we can construct an algorithm via batching with batch length τ to get an algorithm with switching regret*

$$R^{switch,\lambda} \leq \tau R\left(\frac{T}{\tau}, k\right) + \lambda\frac{T}{\tau}.$$

Corollary E.3. *An algorithm with external regret $\mathcal{O}\left(\sqrt{T\log k}\right)$ can be transformed to have switching regret $\mathcal{O}\left(T^{2/3}\sqrt{\log k}\right)$ by choosing $\tau = T^{1/3}$.*

E.3 Tracking regret

The tracking regret framework has been studied to achieve good rewards relative to the baseline action changing p times during the T rounds and is related to adaptive regret.

Herbster and Warmuth [1998] provide an algorithm Fixed-Share that achieves m-segment tracking regret $\mathcal{O}\left(\sqrt{Tm(\log N + \log T)N}\right)$ in the full information setting. Cesa-Bianchi et al. [2012] show that the fixed share algorithm is equivalent to online mirror descent with a particular projection. Theorem 4.1 of Shalev-Shwartz et al. [2012] shows how online mirror descent with bandit information can be implemented without degradation of regret.

This argument is stated in the following Theorem by Daniely et al. [2015].

Theorem E.4 (Restatement of theorem by Daniely et al. [2015])**.** *Fixed-Share algorithm in the bandit feedback setting achieves expected k-segment tracking regret at most $\mathcal{O}\left(\sqrt{Tm(\log N + \log T)N}\right)$.*

F Full Proofs of Results

F.1 Proof of Proposition 4.2

Lemma F.1. *For any distribution z over Ω, the probability $\Pr_{\omega\sim z}[\rho(\sigma(\omega)) = \omega]$ is maximized by a deterministic σ, ρ where ρ is injective.*

Proof. For any distribution p over a space X, we will denote by $p(x)$ for $x \in X$, the probability weight p places on x.

$$\Pr_{\omega\sim z}[\rho(\sigma(\omega)) = \omega] = \sum_{i=1}^{N}\sum_{j=1}^{M} z(\omega_i)\cdot\sigma(\omega_i)(m_j)\cdot\rho(m_j)(w_i)$$

Consider the quantities $\sigma(\omega_i)(m_j)\cdot\rho(m_j)(w_i) \in [0,1]$ for $i \in [M], j \in [N]$ and consider their sum,

$$\begin{aligned}\sum_{i\in[N]}\sum_{j\in[M]}\sigma(\omega_i)(m_j)\cdot\rho(m_j)(w_i) &\le \sum_{i\in[N]}\sum_{j\in[M]}\rho(m_j)(w_i)\\ &= \sum_{j\in[M]}\sum_{i\in[N]}\rho(m_j)(w_i) && \text{(Swapping order of summation)}\\ &= M.\end{aligned}$$

Above we expressed the reconstruction success probability $\Pr_{\omega\sim z}[\rho(\sigma(\omega)) = \omega]$ as $\sum_{i\in[N]}\mu_i z(\omega_i)$ where $\mu_i \in [0,1]$ and $\sum_{i\in[N]}\mu_i \le M$.

Therefore reconstruction probability is at most the sum of the M largest of the N quantities $(z(\omega_i))_{i\in N}$. Let us call the M states with the highest weights according to z by F.

This upper bound of the reconstruction probability is achieved by a deterministic ρ that is a bijection between the M messages and F, and a σ that maps each $\omega \in F$ to the message m such that $\rho(m) = \omega$. □

Proposition 4.2. *There is a $\mathrm{poly}(M, N, T)$ time algorithm for the meta-player in the centralized communication game (Definition 4.1), such that the meta-player's expected regret satisfies $\mathbb{E}[R_T] \le 2\sqrt{MT\log N}$.*

Proof. Communication in this game is an online linear optimization problem in disguise. Any fixed choice of an encoder-decoder pair (σ, ρ) will correctly recover at most M states, since ρ maps to only M states. Let $S(\sigma,\rho) = \{\omega \in \Omega \mid \rho(\sigma(\omega)) = \omega\}$ be the set of states correctly recovered by the pair. Then $|S(\sigma,\rho)| \le M$ for all encoder-decoder pairs σ, ρ.

Nature picks $\omega \in \Omega$ and the meta-player gets a reward of 1 when $\omega \in S(\sigma,\rho)$. For $X \subseteq \Omega$, let $\mathbb{1}_X \in \mathbb{R}^N$ be the indicator vector with 1s on indices corresponding to states in X and 0s elsewhere. The meta-player gets a reward of $\mathbb{1}_{\{\omega\}}^T \mathbb{1}_{S(\sigma,\rho)}$.

More generally, we can think of the problem of picking encoder-decoder pairs as solving a combinatorial optimization problem, where we think of nature as picking a basis element in $e \in \mathbb{R}^N$ and the meta-player as picking a vector $v \in \{0,1\}^N$ that is M-sparse, so $\|v\|_0 = M$. The meta-player gets a reward of $e^T v$.

Of the many algorithms proposed to solve this problem, the Follow the Perturbed Leader algorithm will be most useful to us, achieving a regret of $2\sqrt{2MT}$ in expectation, with only $\sqrt{2MT}$ switches in expectation. □

Corollary F.2. *There is a* $\text{poly}(M, N, T)$ *time algorithm for the meta-player in the centralized communication game, such that the expected regret satisfies, for any* $\alpha \geq 1$ *that could depend on* M, N, T*:*

$$\mathbb{E}[R_T] \leq 2\alpha\sqrt{2MT}$$

and expected number of switches is at most,

$$\frac{1}{\alpha}\sqrt{2MT}.$$

Proof. We can use a standard mini-batching technique, e.g. like those used in Arora et al. [2012], Altschuler and Talwar [2018], to obtain a more fine-grained control over the tradeoff between our algorithm's regret and its number of switches.

Specifically, batch the T iterations into $\frac{T}{\alpha^2}$ groups of α^2, where we commit to each action for α^2 steps. Now, run the algorithm above with number of iterations T/α^2, and repeat every action α^2 times. Because the adversary is oblivious, this cannot increase regret by more than a factor of α^2, resulting in regret

$$2\alpha^2\sqrt{2M(T/\alpha^2)} = 2\alpha\sqrt{2MT},$$

but with an expected number of switches of at most

$$\sqrt{2M(T/\alpha^2)} = \frac{1}{\alpha}\sqrt{2MT}.$$ □

F.2 Proof of Theorem 5.3

Lemma F.3. *There is a* $\text{poly}(M, N, T)$ *time algorithm for the meta-player in the centralized communication game, such that the expected regret satisfies, for any* $\alpha \geq 1$ *that could depend on* M, N, T*:*

$$\mathbb{E}[R_T] \leq 4\alpha\sqrt{TM(1+\log(N))} + 4M(1+\log(N)) \in \mathcal{O}\left(\alpha\sqrt{TM\log(N)} + M\log(N)\right)$$

where for any $\delta \in (0, 1]$*, the number of switches is at most, with probability at least* $1 - \delta$*:*

$$\frac{1}{\alpha}\sqrt{TM(1+\log(N))\log(1/\delta)} \in \mathcal{O}\left(\frac{1}{\alpha}\sqrt{TM\log(N)\log(1/\delta)}\right).$$

Proof. The Multiplicative Follow the Lazy Leader (FTLL*) algorithm will allow us to prove high probability bounds on the number of switches.

For the sake of completeness, we present the algorithm from Kalai and Vempala [2005] below:

Algorithm 2 The Multiplicative Follow the Lazy Leader Algorithm (FTLL*) [Kalai and Vempala, 2005]

1: Choose an initial perturbation $p^{(1)}$ sampled from the Laplace distribution $q(x) \propto e^{-\epsilon|x|_1}$.
2: Let $S_i^{(t)}$ be the ongoing sum of utility for each action i on step t.
3: **for** $t \leftarrow 1$ to T **do**
4: Pick the encoding scheme that maximizes $S_i^{(t)} + p_i^{(t)}$, and receive the vector of rewards $r^{(t)}$. For the communication game, this is an indicator vector for the state the adversary picks.
5: Switch with probability $\max\left(0, 1 - \frac{q(p^{(t)} - r^{(t)})}{q(p^{(t)})}\right)$, setting $p_{t+1} = p_t$.
6: Otherwise, don't switch your expert, so set $p^{(t+1)} = p^{(t)} - r^{(t)}$.
7: **end for**

By Kalai and Vempala [2005, Theorem 1.1, Lemma 1.2], this achieves a regret of

$$2\epsilon T + \frac{2M(1+\log(N))}{\epsilon} + 4M(1+\log(N)).$$

Setting

$$\epsilon = \sqrt{\frac{M(1+\log N)}{T}},$$

this achieves a regret of at most, in expectation,

$$4\sqrt{TM(1+\log N)} + 4M(1+\log(N)).$$

To prove a high probability bound on switching, let X_t be the number of switches FTLL* performs up to iteration t, and let $Y_t = X_t - \epsilon t$. Then, for any t, we can see that this sequence increases by at most one at each iteration: $|Y_{t+1} - Y_t| \leq 1$.

Notice that the probability of switching at each step does depend on whether we have switched on previous steps. Regardless, as shown in the proof of Kalai and Vempala [2005, Lemma 1.2], the probability of switching is always at most ϵ. Indeed, we always use fresh randomness at each step to decide to switch, and exactly as Kalai and Vempala [2005] argue, they do so with probability at most on step t:

$$\begin{aligned}
1 - \frac{q(p^{(t)} - r^{(t)})}{q(p^{(t)})} &= 1 - \exp\left(-\epsilon(\left|p^{(t)} + r^{(t)}\right|_1 - \left|p^{(t)}\right|_1\right) \\
&\leq 1 - \exp\left(-\epsilon\left|r_1^{(t)}\right|\right) \\
&\leq \epsilon\left|r^{(t)}\right|_1 = \epsilon.
\end{aligned}$$

Therefore, $\mathbb{E}[X_{t+1} \mid X_t] \leq \epsilon + X_t$. And so, $\mathbb{E}[Y_{t+1} \mid Y_t] = \mathbb{E}[X_{t+1} \mid Y_t] - \epsilon(t+1) \leq \mathbb{E}[X_t \mid Y_t] + \epsilon t = Y_t$, proving that the sequence of Y_t form a super-martingale.

By Azuma's inequality, this means that:

$$\mathbb{P}\left(X_T > \epsilon T + \sqrt{T\log(1/\delta)}\right) < \delta.$$

We have shown that with probability at least $1-\delta$, the Multiplicative Follow the Lazy Leader algorithm must change at most

$$\sqrt{TM(1+\log(N))} + \sqrt{T\log(1/\delta)}$$

times.

To be able to scale by α, we can use the same mini-batching argument as in Corollary F.2. This results in regret:

$$4\alpha^2\sqrt{(T/\alpha^2)M(1+\log(N))} + 4M(1+\log(N)) = 4\alpha\sqrt{TM(1+\log(N))} + 4M(1+\log(N)).$$

But now with a factor of α less switches:

$$\sqrt{(T/\alpha^2)M(1+\log(N))\log(1/\delta)} = \frac{1}{\alpha}\sqrt{TM(1+\log(N))\log(1/\delta)}. \qquad \square$$

Theorem 5.3. *Suppose the sender and receiver follow the stable sender, adaptive receiver protocol (Definition 5.2) with the algorithms $\mathcal{A}_{\text{stable}}, \mathcal{A}_{\text{adaptive}}$, then the communication regret is at most*

$$R_T \leq R_T^{ext}(\mathcal{A}_{\text{stable}}) + R_T^{track}(\mathcal{A}_{\text{adaptive}}; S_T(\mathcal{A}_{\text{stable}}; \delta)) + \delta T,$$

for every $\delta > 0$, where $S_T(\mathcal{A}_{\text{stable}}; \delta)$ is the number of switches made by $\mathcal{A}_{\text{stable}}$ with probability at least $1-\delta$ and R_T^{ext} and R_T^{track} denote external and tracking regrets.

There exist efficient algorithms $\mathcal{A}_{\text{stable}}, \mathcal{A}_{\text{adaptive}}$ that result in expected communication regret

$$\mathbb{E}[R_T] \in \mathcal{O}\left(MT^{2/3}(N\log N)^{1/3}(\log T)^{1/6} + M\log N\right).$$

Proof. Let $(\overline{\sigma}_t, \overline{\rho}_t)$ be the sequence output by algorithm $\mathcal{A}_{\text{stable}}$ and let ρ_t be the sequence output by $\mathcal{A}_{\text{adaptive}}$. The sequence of strategies the sender and receiver employ according to the communication protocol is $(\overline{\sigma}_t, \rho_t)$.

We can decompose the communication regret of $(\overline{\sigma}_t, \rho_t)$ into two parts: 1) the sub-optimality of $(\overline{\sigma}_t, \overline{\rho}_t)$ and 2) the difference between $(\overline{\rho}_t)$ and (ρ_t). The parts correspond to the sub-optimality in the compression problem and lack of synchronization between the sender and the receiver respectively.

$$\begin{aligned}
R_T\left((\overline{\sigma}_t, \rho_t)_{t=1}^T\right) &= \max_{\sigma,\rho} \sum_{t=1}^T r_t((\sigma, \rho)) - \sum_{t=1}^T r_t((\overline{\sigma}_t, \rho_t)) \\
&= R_{\text{comp}} + R_{\text{sync}}, \text{ where,} \\
R_{\text{comp}} &= \max_{\sigma,\rho} \sum_{t=1}^T r_t((\sigma, \rho)) - \sum_{t=1}^T r_t((\overline{\sigma}_t, \overline{\rho}_t)) \\
R_{\text{sync}} &= \sum_{t=1}^T r_t((\overline{\sigma}_t, \overline{\rho}_t)) - \sum_{t=1}^T r_t((\overline{\sigma}_t, \rho_t))
\end{aligned}$$

Observe that R_{comp} measures the external regret of the output of $\mathcal{A}_{\text{stable}}$ which is at most the switching regret. Hence $R_{\text{comp}} \leq R_T^{\text{switch}}(\mathcal{A}_{\text{stable}})$.

Let p be the number of times the sequence $(\overline{\sigma}_t, \overline{\rho}_t)_{t=1}^T$ switches. We know $p \leq R_T^{\text{switch}}(\mathcal{A}_{\text{stable}})$ Let $s_1, \ldots, s_p$ be the time indices of the switches. Then,

$$\begin{aligned}
R_{\text{sync}} &= \sum_{t=1}^T r_t((\overline{\sigma}_t, \overline{\rho}_t)) - \sum_{t=1}^T r_t((\overline{\sigma_t}, \rho_t)) \\
&= \sum_{i=1}^p \sum_{t=s_i}^{s_{i+1}} r_t(\overline{\sigma}_{s_i}, \overline{\rho}_{s_i}) - \sum_{t=1}^T r_t((\sigma_t, \rho_t)) \\
&\leq \max_{\substack{s_1,\ldots,s_p \\ \sigma_1,\ldots,\sigma_p \\ \rho_1,\ldots,\rho_p}} \sum_{i=1}^p \sum_{t=s_i}^{s_{i+1}} r_t(\sigma_i, \rho_i) - \sum_{t=1}^T r_t((\sigma_t, \rho_t)) \\
&\leq R_T^{\text{track}}(p) \leq R_T^{\text{track}}(\mathcal{A}_{\text{adaptive}}; R_T^{\text{switch}}(\mathcal{A}_{\text{stable}})).
\end{aligned}$$

Now we move to the second part of the proof which is to show the existence of $\mathcal{A}_{\text{stable}}, \mathcal{A}_{\text{adaptive}}$ that result in the communication regret bound of the theorem statement. We mainly draw on results from previous work in the switching regret and tracking regret frameworks to do this.

Here, we describe $\mathcal{A}_{\text{stable}}, \mathcal{A}_{\text{adaptive}}$, state their switching regret and tracking regret guarantees, and show how this results in the communication regret bound in the theorem.

The algorithm $\mathcal{A}_{\text{stable}}$ needs to minimize switching regret over the space of all encoder-decoder pairs. Recall that in our warm up study of the centralized communication setting, we already have an algorithm $\mathcal{A}$ that minimizes the external regret over this space. In Lemma F.3, we showed that the Multiplicative Follow the Lazy Leader algorithm has a small number of switches with high probability. Setting the mini-batching into $\alpha = \frac{T^{1/6} M^{1/2} N^{1/3} (\log T)^{1/6}}{(\log N)^{1/6}}$ batches and using $\delta = 1/T$, we get an expected regret of $\mathcal{O}\left(MT^{2/3}(N \log N)^{1/3}(\log T)^{1/6} + M \log(N)\right)$. With probability at least $1 - 1/T$, this performs at most $\mathcal{O}\left(T^{1/3}(\log N)^{2/3}(N \log T)^{1/6}\right)$ switches.

Next, to construct $\mathcal{A}_{\text{adaptive}}$ that minimizes tracking regret with $\mathcal{O}\left(T^{2/3}(\log N)^{2/3}(N \log T)^{1/6}\right)$ segments over the space of decoders, we use a standard tracking regret minimizing algorithm $\overline{\mathcal{A}}_{\text{adaptive}}$ such as AdaNormalHedge Luo and Schapire [2015] or the Fixed share algorithm Herbster and Warmuth [1998], Cesa-Bianchi et al. [2012] as a base. We create M copies of this algorithm $(\overline{\mathcal{A}}_{\text{adaptive}}^{(m)})$ associated with every message in $\mathcal{M}$.

When $\mathcal{A}_{\text{adaptive}}$ sees a message m, it outputs the state output by the associated algorithm $\overline{\mathcal{A}}_{\text{adaptive}}^{(m)}$ and updates this copy leaving all other copies unchanged.

Each copy $\overline{\mathcal{A}}^{(m)}_{\text{adaptive}}$ achieves expected $S_T = \mathcal{O}\left(T^{1/3}(\log N)^{2/3}(N \log T)^{1/6}\right)$ segments tracking regret at most $\mathcal{O}\left(\sqrt{TN \cdot S_T}\right) \in \mathcal{O}\left(T^{2/3}(N \log N)^{1/3}(\log T)^{1/6}\right)$. The tracking regret of $\mathcal{A}_{\text{adaptive}}$ is at most the sum of tracking regrets of the copies and hence can be bounded by $\mathcal{O}\left(MT^{2/3}(N \log N)^{1/3}(\log T)^{1/6} + M \log(N)\right)$. □

F.3 Proof of Theorem 5.4

Theorem 5.4. *(Synchronization with Initial Setup) Given any online algorithm $\mathcal{A}$ over the space $\Sigma \times \mathrm{P}$ of encoder-decoder pairs, there are sender and receiver algorithms that achieve communication regret at most*

$$R_T \leq R_T^{ext}(\mathcal{A}) + M \log N \cdot S_T(\mathcal{A}; \delta) + \delta T,$$

for any $\delta > 0$, where $S_T(\mathcal{A}; \delta)$ is the number of switches made by $\mathcal{A}$ with probability at least $1 - \delta$.

There are efficient sender and receiver algorithms that result in expected regret

$$\mathbb{E}[R_T] \in \mathcal{O}\left(T^{1/2} M (\log N)^{1/2}\right).$$

Proof. The sender uses $\mathcal{A}$ in mostly the same way as in the previous protocol (Definition 5.2), generating encoder-decoder pairs and employing the encoder of this pair in the communication game. However, instead of always employing the encoder from $\mathcal{A}$, the sender uses some rounds after switching encoders to explicitly communicate the new decoder the receiver should use to the receiver.

Since there are N^M deterministic decoders (all mappings from $[N]$ to $[M]$), the sender can communicate the new decoder to the receiver in $\log_M(N^M) = M \log_M(N)$ steps by assigning a unique sequence of messages to each decoder. This protocol crucially assumes that both the sender and receiver share a common mapping of sequence of messages to decoders.

Instead of communicating after every switch in encoder, the sender communicates after the first mistake made by the receiver after the switch. This is to ensure the receiver knows which rounds the sender is using to communicate the new decoder. A mistake after previous synchronization is a common signal for both the sender and receiver to start the meta-communication of communicating the new decoder.

After the completion of the meta-communication of the new decoder, the receiver exactly knows the decoder to use and becomes perfectly synchronized with the sender without any additional exploration. The cost of the synchronization is simply the $M \log_M(N)$ rounds used for the meta-communication.

From here, the regret bound is easy to show with the Corollary F.2 in hand. Every switch results in a regret of $M \log(N)$. Therefore, for any choice of α, we can derive an algorithm with regret

$$2\alpha\sqrt{2MT} + \frac{1}{\alpha}\sqrt{2MT} M \log_M(N).$$

Picking $\alpha = \sqrt{2M \log_M(N)}$ gives a final regret bound of at most

$$4M\sqrt{T \log_M(N)},$$

as claimed. □

F.4 Proof of Proposition 5.5

Proposition 5.5. *(Synchronization in plain mode) Given any online algorithm $\mathcal{A}$ over the space $\Sigma \times \mathrm{P}$ of encoder-decoder pairs, there are sender and receiver algorithms that don't rely on initial shared meaning of messages that achieve communication regret at most*

$$R_T \leq R_T^{ext}(\mathcal{A}) + \mathcal{O}\left(MN^3 \log(1/\delta)\right) \cdot S_T(\mathcal{A}),$$

with probability at least $1 - \delta$ for $\delta > 0$, where $S_T(\mathcal{A})$ is the number of switches made by $\mathcal{A}$.

There are efficient sender and receiver algorithms that don't rely on initial shared meanings of messages that result in expected communication regret

$$\mathbb{E}[R_T] \in \mathcal{O}\left(T^{1/2} M N^{3/2} (\log N)^{1/2}\right).$$

Proof. Receiver's protocol. The receiver's protocol consists of three components: 1) an exploration phase, 2) an exploitation phase, and 3) detection of change. The receiver starts off in the exploration phase and then performs exploitation until a shift is detected at which point the receiver changes back to exploration.

Receiver's exploration. At the start of exploration, the receiver maps all messages to $\perp$. For each message, until the mapping remains $\perp$, the receiver iterates through all possible states in a random ordering until decoding to a state results in a reward. At this point, the receiver changes the mapping of the message to the state that yielded the reward.

When the receiver has all messages mapped to a state instead of $\perp$, the receiver continues to use this decoder until a shift is detected. A shift is detected in the following way.

Shift detection. The receiver's shift detection is designed accounting for the following property of the sender's protocol (we will describe the full protocol later). The sender always employs an encoder such that the pre-image set of $M-1$ messages has size 1 and only one message has a pre-image containing multiple states.

This means that the synchronized decoder will only ever make mistakes when decoding one message. If the decoder makes makes mistakes on two different messages, this means that the decoder is no longer synchronized with the sender's encoder and a shift in the sender's encoder is detected moving the receiver back into exploration.

Sender's protocol. The sender's algorithm minimizes switching regret over the space of encoder-decoder pairs as in the protocol defined by Definition 5.2 to obtain a sequence of pairs (σ_t, ρ_t).

At round t, the sender plays the encoder $\mathrm{BR}(\rho_t)$ that is optimal to ρ_t and maps $M-1$ states in $\mathrm{Im}(\rho)$ to messages $1, \ldots, M-1$ and all other states to message M.

If in a round t with $\omega_t^* \notin \mathrm{Im}(\rho_t)$, the sender receives a reward, in the next two rounds with distinct realized states in $\mathrm{Im}(\rho_t)$, the sender intentionally causes a mistake to reset the receiver.

Regret analysis. Suppose $(\overline{\sigma}_t, \overline{\rho}_t)$ are the sequence of strategies output by the sender's stable algorithm with $O(T^{2/3})$ switches.

For each distinct encoder-pair $(\overline{\sigma}^{(i)}, \overline{\rho}^{(i)})$, we will show that with high probability, within $O(MN^2)$ mistakes, the receiver will start choosing the strategy.

For each of the messages m from m_1 to m_{M-1}, the receiver will find $\overline{\rho}^{(i)}(m)$ after making at most N mistakes.

The tricky part is analyzing how the receiver finds the mapping $\overline{\rho}^{(i)}(m_M)$ when multiple states are mapped to m_M by $\overline{\sigma}^{(i)}$.

In fact, the receiver might incorrectly form a mapping that differs from $\overline{\rho}^{(i)}(m_M)$. However, the sender will force a mistake by intentionally using a different encoding for a message in $m_1, \ldots, m_{M-1}$.

We will now bound the number of times the sender would cause this intentional mistake to reset the receiver. A sufficient condition for the sender to stop resetting is that when the catch-all message is generated, the intended state for this message is generated and the receiver guesses the intended state. The probability of this is at least $1/N^2$. Therefore with probability at least $1-\delta$, the number of resets is $\mathcal{O}\left(N^2 \log(1/\delta)\right)$.

We saw that the number of mistakes at each reset is $\mathcal{O}(MN)$. The regret due to lack of synchronization is at most $\mathcal{O}\left(MN^3 S_T \log T\right)$, where S_T is the number of switches of $\mathcal{A}$. We know by Corollary F.2 that, for any choice of $\alpha \geq 1$, there is a $\mathcal{A}$ with external regret $2\alpha\sqrt{TM \log N}$ and with expected regret $1/\alpha\sqrt{2MT}$ many switches. Set $\alpha = \sqrt{MN^3}$. Then, the regret of this procedure is $\mathcal{O}\left(T^{1/2} M N^{3/2} (\log N)^{1/2}\right)$. □

F.5 Proof of Proposition B.1

Proposition B.1. *There is an efficient algorithm that allows the sender and receiver to learn a language with regret* $\mathbb{E}[R_T] \in \tilde{\mathcal{O}}\left(MT^{1/2}(\log N)^{1/2}\right)$ *in the centralized communication game that switches at most* $1 + \log\log T$ *times.*

Proof. We present a method that achieves a regret of $T^{1/2}$ that only requires the sender to switch their strategy $\mathcal{O}(\log\log T)$ times. This is inspired by Cesa-Bianchi et al. [2013, 2014].

The key idea is to have a sequence of stages where stage s lasts for T_s steps, starting with $s = 0$:

1. Use the empirical estimate of the probability of each state so far to estimate the optimal communication strategy.

2. Commit to this for T_s steps, all the while gathering more data on the number of times each state appears.

3. After T_s steps have run, increment s and go back to step 1 if we haven't run for at least T steps yet.

Set the length of each stage to be $T_s = T^{1-2^{-s}}$. We can show that this is run for at most $1 + \log_2 \log_2 T$ stages. Indeed, the length of just the last two stages are:

$$\begin{aligned} T^{1-2^{-(1+\log_2 \log_2 T)}} + T^{1-2^{-\log_2 \log_2 (T)}} &\geq 2T^{1-2^{-\log_2\left(\frac{\log_2(T)}{\log_2(M)}\right)}} \\ &= T \cdot 2T^{-\frac{1}{\log_2(T)}} \\ &= T \cdot 2 \cdot 2^{-\log_2 T \cdot \frac{1}{\log_2(T)}} \\ &\geq T \cdot 2 \cdot \frac{1}{2} = T. \end{aligned}$$

How good is our estimate at this stage? Lemma F.4 gives us an answer to this.

Lemma F.4. *With access to α draws of the underlying distribution of the stochastic communication game, the sender and receiver can coordinate on a policy in the centralized game that achieves regret*

$$\frac{1}{\sqrt{2\alpha}} \cdot \left(2 + M\sqrt{\log(N\sqrt{\alpha})}\right) \in \tilde{\mathcal{O}}\left(\frac{M}{\sqrt{\alpha}}\right).$$

per step.

Proof. Let $p(\omega)$ be the probability that $\omega \in \Omega$ is drawn from the distribution. If we have access to $\hat{p}(\omega)$ such that $|\hat{p}(\omega) - p(\omega)| < \tau$, the encoding scheme corresponding to Lemma F.1 achieves $M\tau$ regret per iteration. Indeed, let (σ^*, ρ^*) be the optimal deterministic encoder-decoder pair under the true distribution of states p. For each state $\omega \in \Omega$, $\rho^*(\sigma^*(\omega))$ is mapped to a deterministic state, and in particular, the agents only get utility when $\omega = \rho^*(\sigma^*(\omega))$. Because ρ^* only has M possible inputs, it can only have at most M possible outputs, and so there can only be at most M states $\omega_{i_1}, \ldots, \omega_{i_k}$ such that $\rho^*(\sigma^*(\omega_{i_j})) = \omega_{i_j}$. Therefore, the utility this scheme gets is precisely $\sum_{j=1}^{k} p(\omega_{i_j})$. When the underlying distribution is the approximate $\hat{p}$, since $\left|p(\omega_{i_j}) - \hat{p}(\omega_{i_j})\right| < \tau$, the utility cannot differ by more than $\left|\sum_{j=1}^{k} p(\omega_{i_j}) - \sum_{j=1}^{k} \hat{p}(\omega_{i_j})\right| \leq k\tau \leq M\tau$. Therefore, the optimal encoder-decoder scheme under probabilities $\hat{p}$ gets a utility at least $M\tau$ close to the true optimal utility.

Let S_t be the state drawn from the hidden distribution at time t, and let $X_{\omega,t}$ be the random variable that is 1 when $S_t = \omega$. After receiving α samples of the distribution, for any state ω, the probability that the empirical estimate of the state distributions is at least $\tau = \frac{1}{\sqrt{2\alpha}}\sqrt{\log(2N/\delta)}$ away from the true expectation by Hoeffding's inequality is:

$$\mathbb{P}\left(\left|\frac{\sum_{t=1}^{\alpha} X_{\omega,t}}{\alpha} - p_\omega\right| > \tau\right) \leq 2\exp\left(-\frac{2(\alpha\tau)^2}{\alpha}\right) = \frac{\delta}{N}.$$

By the union bound, the probability that the estimates for all N states are within τ is then $1 - \delta$. In this case, as shown above, we achieve a regret of τ per step. Because the probability of failure is at most δ, in expectation, regret per step is at most $\delta + M\tau$. Setting $\delta = 2/\sqrt{\alpha}$, we get a regret of at most

$$\frac{2}{\sqrt{2\alpha}} + \frac{M}{\sqrt{2\alpha}}\sqrt{\log(N\sqrt{\alpha})}$$

per time step. □

By the time we have reached stage s, the receiver has available to them at least $T^{1-2^{-s+1}}$ samples from the previous stage. This stage is run for $T^{1-2^{-s}}$ steps. By Lemma F.4, this step then receives a regret of at most

$$\frac{T^{1-2^{-s}}}{\sqrt{2T^{1-2^{-s+1}}}} \cdot \left(2 + M\sqrt{\log\left(N\sqrt{T^{1-2^{-s+1}}}\right)}\right) \leq \frac{1}{\sqrt{2}} \cdot T^{1/2}\left(2 + M\sqrt{\log(NT)}\right).$$

Summing over at most all $1 + \log\log T$ stages, we achieve the claimed regret of

$$\frac{1}{\sqrt{2}} \cdot T^{1/2}\left(2 + M\sqrt{\log(NT)}\right)(1 + \log\log T). \qquad \square$$

F.6 Proof of Corollary C.1

Corollary C.1. *Suppose the sender and receiver follow the stable sender, adaptive receiver protocol in a communication game with general utilities (Definition 5.2) using the algorithms $\mathcal{A}_{\text{stable}}, \mathcal{A}_{\text{adaptive}}$, then the communication regret is at most*

$$R_T \leq R_T^{ext}(\mathcal{A}_{\text{stable}}) + R_T^{track}(\mathcal{A}_{\text{adaptive}}; S_T(\mathcal{A}_{\text{stable}}; \delta)) + \delta T,$$

for every $\delta > 0$, where $S_T(\mathcal{A}_{\text{stable}}; \delta)$ is the number of switches made by $\mathcal{A}_{\text{stable}}$ with probability at least $1 - \delta$ and R_T^{ext} and R_T^{track} denote external and tracking regrets.

There exist efficient algorithms $\mathcal{A}_{\text{stable}}, \mathcal{A}_{\text{adaptive}}$ that result in expected $(1 - 1/e)$-communication regret

$$\mathbb{E}[R_T] \in \mathcal{O}\left(ST^{2/3}M^{4/3}N^{1/3}\log(N)^{1/3}\log(MT)^{1/6} + SM^2\log(N)\right).$$

Proof. This will combine Theorem 5.3 with Theorem 6.2. The first half follows directly from the argument in Theorem 5.3. For the second half, we will use Algorithm 1, but instead use M copies of the no-switching regret Algorithm 2.

Using Lemma F.3, we can set $\delta = 1/MT$ and $\alpha = T^{1/6}N^{1/3}M^{-1/6}\log(N)^{-1/6}\log(MT)^{1/6}$, resulting in each copy of Algorithm 2 having regret at most $M\log(N) + T^{2/3}M^{1/3}\log(N)^{1/3}N^{1/3}\log(MT)^{1/6}$ and switching at most $T^{1/3}M^{2/3}N^{1/6}\log(N)^{2/3}\log(MT)^{1/3}$ times with probability at least $1 - 1/MT$. By the union bound, with probability at least $1 - 1/T$, all the copies will switch at most $T^{1/3}M^{2/3}N^{1/6}\log(N)^{2/3}\log(MT)^{1/3}$ times each. By [Streeter and Golovin, 2008, Lemma 3], we can bound the expected $(1 - 1/e)$-regret by the sum of the regrets of each copy of FTLL*, which results in a regret of at most $M^2\log(N) + T^{2/3}M^{4/3}\log(N)^{1/3}N^{1/3}\log(MT)^{1/6}$.

Just as in Theorem 5.3, we can use M copies of a standard tracking regret minimization algorithm. With probability $1/T$, the number of switches for each copy will be at most $S_T = T^{1/3}N^{1/6}M^{2/3}\log(N)^{2/3}\log(MT)^{1/3}$, so will achieve a regret of at most $\sqrt{TNS_T} \in \mathcal{O}\left(T^{2/3}N^{1/3}M^{1/3}\log(N)^{1/3}\log(MT)^{1/6}\right)$. But we have M copies of the adaptive algorithm, so total regret becomes $\mathcal{O}\left(T^{2/3}N^{1/3}M^{4/3}\log(N)^{1/3}\log(MT)^{1/6}\right)$.

Adding the regret of all the sender and receiver together, we achieve a total $(1 - 1/e)$-regret of at most

$$\mathcal{O}\left(T^{2/3}N^{1/3}M^{4/3}\log(N)^{1/3}\log(MT)^{1/6} + M^2\log(N)\right).$$

Scaling by S, we get our final answer. $\square$

F.7 Proof of Theorem B.3

Theorem B.3. *There is an efficient no-regret algorithm that can achieve a $(1 - 1/e)$-approximation of optimal communication with an arbitrary utility $r : \Omega \times \mathcal{A} \to [0, U]$ in stochastic environments with regret $\tilde{\mathcal{O}}\left(UMT^{1/2}\right)$.*

Proof. Simply run the algorithm presented in Proposition B.1. Except, to find the messaging scheme that maximizes expected utility computed using the current estimate of probabilities of each state $\hat{p}$, by Theorem 6.2, we must find the set of M actions that maximizes:

$$V(\{a_1, \ldots, a_M\}) = \mathop{\mathbb{E}}_{\omega \sim \hat{p}}\left[\max_i r(a_i, \omega)\right].$$

We can use the greedy algorithm in Buchbinder et al. [2014][Theorem 3.1] to get a $(1 - 1/e)$-approximation of optimal utility.

When our approximates are within τ of the true state distribution p, i.e. $|p_\omega - \hat{p}_\omega| < \tau$, it must be that the solution we compute is never more than an additive $UM\tau$ away from a $(1 - 1/e)$ approximation of optimal in expected reward. This robustness of the greedy algorithm to additive error has been noted before by Streeter and Golovin [2008, Theorem 6]. Indeed, every step of the greedy algorithm is never off by more than $\tau \max_i r(a_i, \omega) \leq U\tau$, and the algorithm is performed for M steps.

Running the proof as in Proposition B.1 from here, we get a regret of at most

$$\left(\frac{1}{\sqrt{2}} T^{1/2} \left(2 + UM\sqrt{\log(NT)}\right) + M \log_M(A)\right) \cdot (1 + \log\log T),$$

where we use $\log_M(A)$ instead of $\log_M(N)$ in the switching cost because a decoder that maps to actions instead of states is easier to communicate. □

F.8 Proof of Lemma C.2

Lemma C.2. *Computing any stochastic communication strategy that is an α-approximation of the optimal strategy, where $\alpha > 1 - 1/e$, in a communication game with rewards restricted to be 0 or 1 is NP-hard.*

Proof. Recall that the problem of finding α-approximations to the max k-coverage problem with $\alpha > 1 - 1/e$ is NP-hard [Feige, 1998]. Then, we will reduce this to finding an α-approximation of optimal communication in a communication game. The utilities in this game will be 0 or 1 only.

Given a value k and a collection of sets $S = \{S_1, \ldots, S_m\}$ each with elements in a universe $\mathcal{U}$, Feige [1998] proved that it is NP-hard to get α-approximations to this problem.

Create a communication game with $M = k$ messages over the state space $\mathcal{U}$. Make each S_i an action, and use the following reward function:

$$r(e, S_i) = \begin{cases} 1 & \text{if } e \in S_i, \\ 0 & \text{otherwise.} \end{cases}$$

Let the underlying distribution $\mathcal{D}$ assign an equal probability to each state, $1/|\mathcal{U}|$.

We will prove two useful facts about this game.

1. Every solution to the max-M-coverage problem induces a solution to this communication game with the same value. More precisely, every collection of sets $S_{i_1}, \ldots, S_{i_M}$, with union $S = \bigcup_{j=1}^{M} S_{i_j}$, induces an encoder-decoder pair σ, ρ that achieves utility that is at least the value of the sets in the maximum-M-coverage problem: $|S| / |\mathcal{U}|$. Indeed, let the sender sends message m_j when observing a state in $S_{i_j} \setminus \bigcup_{\ell=1}^{j-1} S_{i_\ell}$. The receiver plays set S_{i_j} upon receiving message m_j. Let $S = \bigcup_{j=1}^{M} S_i$. Whenever a state in S appears, the agents get a reward of 1, and so the expected reward of this encoder-decoder pair is $|S| / |\mathcal{U}|$, precisely the value of the sets in the maximum-M-coverage problem.

2. Every solution with a deterministic decoder to this communication game induces a solution to the max-M-coverage problem with at least the same value. More precisely, given a deterministic decoder ρ, we will construct a collection of sets $S_{i_1}, \ldots, S_{i_M}$, with union $S = \bigcup_{j=1}^{M} S_{i_j}$, so that for any encoder σ, the utility ρ, σ achieves utility is at most the value of the sets in the maximum-M-coverage problem: $|S| / |\mathcal{U}|$.

   ρ maps each message to an action S_i. There are M possible messages, so simply take the M actions, $S_{i_1}, \ldots, S_{i_M}$, that the receiver chooses to play as the solution to the maximum coverage problem. Let $S = \bigcup_{j=1}^{M} S_{i_j}$. The value ρ achieves with any encoder σ is at most the probability that elements in S appear: $|S| / |\mathcal{U}|$.

These reductions back and forth imply that the optimal values of both problems are equal.

In general communication games, given an encoder σ that isn't necessarily deterministic, there exists a deterministic best-response decoder ρ^* that can be found by an efficient algorithm. Indeed, given message $m \in \mathcal{M}$, the value of playing action $a \in \mathcal{A}$ is

$$\begin{aligned}\mathop{\mathbb{E}}_{\omega}[r(a,\omega) \mid \sigma(\omega) = m] &= \sum_{\omega \in \Omega} \mathbb{P}(\omega \mid \rho(\omega) = m) r(a,\omega) \\ &= \sum_{\omega \in \Omega} \frac{\mathbb{P}(\omega)\mathbb{P}(\sigma(\omega) = m)}{\sum_{\omega' \in \Omega} \mathbb{P}(\omega')\mathbb{P}(\sigma(\omega') = m)} r(a,\omega),\end{aligned}$$

which can be computed in polynomial time in the number of actions and states. The reward any response achieves is linear, and the optimal deterministically plays the action with highest expected reward given the message received.

Using these two facts, we can reduce the problem of finding an α-approximation of the maximum k coverage problem to finding an α-approximation of the optimal encoder-decoder pair in this communication game. Suppose (σ, ρ) are an encoder-decoder pair that achieve an α-approximation of optimal play in this communication game. (σ, ρ) may not necessarily be deterministic. But, by the above, we can construct in polynomial time a deterministic decoder ρ_{det} that achieves the same reward.

Applying fact 2 on ρ_{det} this means that we have found a collection of sets $S_{i_1}, \ldots, S_{i_k}$ that achieve at least α times the max reward of the communication game. So we have shown that the optimal reward in this game is equal to the optimal reward in the maximum-k-coverage problem, and we are done! □

F.9 Proof of Theorem C.3

Theorem C.3. *Unless RP = NP, for any $\alpha > 1 - 1/e$, any algorithm that runs in time poly(N, M) per iteration has α-approximate communication regret such that either $R_T^\alpha \in \omega(T^{1-\epsilon})$ for all $\epsilon > 0$ or $R_T^\alpha \notin$ poly(N, M).*

Proof. Suppose for the sake of contradiction that $R_T \in \mathcal{O}\left(T^{1-\epsilon}\right)$ for some $\epsilon > 0$.

We can write for some constants $c_1, c_2, a, b \in \mathbb{R}$, that $R_T(N, M) \leq c_1 N^\alpha M^\beta T^{1-\epsilon} + c_2$. This means that the average regret per time step is at most

$$\frac{c_1 N^a M^b}{T^\epsilon} + \frac{c_2}{T}.$$

The rest of the proof will follow the lead of Kapralov et al. [2013]. Given a value k and a collection of sets $S = \{S_1, \ldots, S_m\}$ each with elements in a universe $\mathcal{U}$, Feige [1998] proved that, for any $\delta > 0$, it is NP-hard to distinguish between the following two cases:

1. Yes case: Some choice of k sets covers $\mathcal{U}$.
2. No case: No choice of k sets covers a $1 - 1/e + \delta$ proportion of the elements in $\mathcal{U}$.

Therefore, if there is a polynomial-time algorithm that outputs yes with constant probability when in the yes case, and outputs no always when in the no case, it must be that RP = NP.

Suppose there exists an efficient no-α-approximate regret communication algorithm. We will show that the problem above for $\delta = \frac{\alpha + (1 - 1/e)}{2}$ can be solved with a constant probability of success in the yes case, and always in the no case. This would prove RP = NP.

Given an instance of the set-cover problem, run the algorithm on the corresponding communication game derived in Lemma C.2. As we show in Lemma C.2, the optimal value in this game is equal to that of the set-cover problem.

Run the no-α-approximate regret learning algorithm for

$$T = \max\left(c_2,\ \left(\left(\frac{\alpha - (1 + 1/e)}{4}\right) c_1 N^a M^b\right)^{1/\epsilon}\right) \in \text{poly}(N, M)$$

steps, and the average regret per iteration becomes at most $\frac{\alpha-(1-1/e)}{4}$.

First, notice that we may as well pretend that the sender and receiver output the full ρ_t and σ_t at every step, not just $\sigma_t(\omega_t)$ and $\rho_t(\sigma_t(\omega_t))$. This is because at every step of the learning algorithm, we can create $|\Omega|$ and $|\mathcal{M}|$ copies and test the behavior of the agents on every counterfactual input.

The reduction will be as follows. Run the algorithm on T i.i.d draws $\omega_1, \ldots, \omega_T \sim \mathcal{D}^T$, induce a sequence of encoder-decoder pairs $(\sigma_1, \rho_1), \ldots, (\sigma_T, \rho_T)$, compute the expected reward of each encoder-decoder pair under $\mathcal{D}$ (which we can do in at most $\mathcal{O}(NT)$ time), and output yes if some pair σ_t, ρ_t exists with expected reward at least $1 - \frac{\alpha-(1-1/e)}{2}$, and no otherwise.

When we are in the no case, no solution will have expected reward in the communication game that is at least $1 - \frac{\alpha-(1-1/e)}{2}$, so we will always output no.

Suppose we are in the yes case, so that the optimal utility is 1. We can say

$$\mathop{\mathbb{E}}_{\omega_{1:T}\sim\mathcal{D}^T}\left[\frac{1}{T}\sum_{t=1}^{T} r(\rho_t(\sigma_t(\omega_t)), \omega_t)\right] \geq 1 - \frac{\alpha-(1-1/e)}{4}. \tag{3}$$

To move forward, we will need to remove the dependence on ω_t in the expected reward. Notice that $\omega_t \perp (\sigma_t, \rho_t)$, as ω_t is sampled from $\mathcal{D}$, and σ_t, ρ_t depends only on $\omega_1, \ldots, \omega_{t-1}$, which themselves are independent of ω_t. Therefore, for any t, it must be that

$$\begin{aligned}
\mathop{\mathbb{E}}_{\omega_{1:T}\sim\mathcal{D}^T}[r(\rho_t(\sigma_t(\omega_t)), \omega_t)] &= \mathop{\mathbb{E}}_{\omega_{1:(t-1)}\sim\mathcal{D}^{(t-1)}} \mathop{\mathbb{E}}_{\omega_t\sim\mathcal{D}}[r(\rho_t(\sigma_t(\omega_t)), \omega_t) \mid \omega_1, \ldots, \omega_{t-1}] \\
&= \mathop{\mathbb{E}}_{\omega_{1:(t-1)}\sim\mathcal{D}^{(t-1)}} \mathop{\mathbb{E}}_{\omega\sim\mathcal{D}}[r(\rho_t(\sigma_t(\omega)), \omega)].
\end{aligned}$$

And so by the linearity of expectation,

$$\begin{aligned}
\mathop{\mathbb{E}}_{\omega_{1:T}\sim\mathcal{D}^T}\left[\frac{1}{T}\sum_{t=1}^{T} r(\rho_t(\sigma_t(\omega_t)), \omega_t)\right] &= \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\omega_{1:T}\sim\mathcal{D}^T} r(\rho_t(\sigma_t(\omega_t)), \omega_t) \\
&= \frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\omega_{1:(t-1)}\sim\mathcal{D}^{(t-1)}} \mathop{\mathbb{E}}_{\omega\sim\mathcal{D}}[r(\rho_t(\sigma_t(\omega)), \omega)] \\
&= \mathop{\mathbb{E}}_{\omega_{1:T}\sim\mathcal{D}^T}\left[\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\omega\sim\mathcal{D}}[r(\rho_t(\sigma_t(\omega)), \omega)]\right].
\end{aligned}$$

Plugging this into (3),

$$\mathop{\mathbb{E}}_{\omega_{1:T}\sim\mathcal{D}^T}\left[\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\omega\sim\mathcal{D}}[r(\rho_t(\sigma_t(\omega)), \omega)]\right] \geq 1 - \frac{\alpha-(1-1/e)}{4}.$$

By Hoeffding's inequality, we can say that with probability at least a constant,

$$1 - 2\exp\left(-\frac{\alpha-(1-1/e)}{4}\right),$$

it must be that

$$\frac{1}{T}\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\omega\sim\mathcal{D}}[r(\rho_t(\sigma_t(\omega)), \omega)] \geq 1 - \frac{\alpha-(1-1/e)}{2}.$$

Because it is an average, we can say in this case that there exists some t such that $\mathbb{E}_{\omega\sim\mathcal{D}}[r(\rho_t(\sigma_t(\omega)), \omega)] \geq 1 - \frac{\alpha-(1-1/e)}{2}$, and our reduction will find it.

Thus, we can efficiently determine whether we are in the yes case with constant probability, and we are done! □

Note that while it severely restricts the efficiency of any learning algorithm for the general communication problem under standard complexity theoretic assumptions, Theorem C.3 does not imply that there doesn't exist a no-regret algorithm. In fact, there could still exist an efficient algorithm where $R_T \in \mathcal{O}\left(T/\log T\right)$.