A MINIMALITY OF CHEAP TALK MDPS

A.1 PROOF OF PROPOSITION 1

Proposition 1. *For any Cheap Talk MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \mathcal{M}, f, \mathcal{J} \rangle$, the policy of a **tabular** Victim initialised uniformly along $\mathcal{M}$ is independent from its Adversary.*

Proof. In a Cheap Talk MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \mathcal{M}, f, \mathcal{J} \rangle$, a tabular Victim arbitrarily orders states as $\{s_1, \dots, s_d\}$ and messages as $\{m_1, \dots, m_k\}$, where $d = |\mathcal{S}|$ and $k = |\mathcal{M}|$, and stores policies $\pi_t(\cdot \mid s_i, m_j)$ at time t of the learning process for all $i \in [d], j \in [k]$. The argument follows identically for value functions. Assuming uniform initialisation along the $\mathcal{M}$ axis means that

$$\pi_0(\cdot \mid s_i, m_j) = \pi_0(\cdot \mid s_i, m_{j'})$$

for all $j, j' \in [k]$. Now consider any two Adversaries f, g and their influence on two copies of the same Victim V, W with respective policies π, χ. The only states encountered in the environment are of the form $(s, f(s))$ and $(s, g(s))$ respectively, so Victims only update the corresponding policies

$$\pi_t(\cdot \mid s_i, f(s_i)) \qquad \text{and} \qquad \chi_t(\cdot \mid s_i, g(s_i))\,.$$

We prove by induction that these quantities are equal for all t. The base case holds by uniform initialisation along $\mathcal{M}$; assume the claim holds for all fixed $0 \leq t \leq T$. The Victims update their policies at time $T+1$ according to the same learning rule, as a function of the transitions and returns under current and past policies π_t and χ_t respectively. Transitions take the form $(s, f(s), a, s', f(s'))$ for V and $(s, g(s), a, s', g(s))$ for W, which have identical probabilities and returns because

$$\begin{aligned}
\pi_t(a \mid s_i, f(s_i)) &= \chi_t(a \mid s_i, g(s_i))\,; \\
\mathcal{P}(s', f(s') \mid s, f(s), a) &= \mathcal{P}(s', g(s') \mid s, g(s), a)\,; \\
\mathcal{R}(s, f(s), a) &= \mathcal{R}(s, g(s), a)
\end{aligned}$$

by inductive assumption and independence of $\mathcal{P}, \mathcal{R}$ from $\mathcal{M}$. This implies that the Victims' policies $\pi_T(\cdot \mid s_i, f(s_i)) = \chi_T(\cdot \mid s_i, g(s_i))$ are updated identically to

$$\pi_{T+1}(\cdot \mid s_i, f(s_i)) = \chi_{T+1}(\cdot \mid s_i, g(s_i))$$

as required to complete induction. Note that this would not necessarily hold in non-tabular settings, where updating parameters θ of the function approximator for some state $(s_i, f(s_i))$ may alter the policy on some other state $(s_j, f(s_j))$. It now follows that trajectories $\tau = (s^k, f(s^k), a^k)_k$ for V and $\omega = (s^k, g(s^k), a^k)_k$ for W have identical probabilities and hence produce identical returns

$$\mathbb{E}_{\tau \sim \pi_t}\left[\mathcal{R}(\tau)\right] = \mathbb{E}_{\omega \sim \chi_t}\left[\mathcal{R}(\omega)\right]$$

at any timestep t of the learning process, concluding independence from Adversaries. □

A.2 PROOF OF PROPOSITION 2

Proposition 2. *A Victim which is **guaranteed to converge to optimal policies in MDPs** will also converge to optimal policies in Cheap Talk MDPs, with an expected return equal to the optimal return for the corresponding no-channel MDP.*

Proof. By assumption, the Victim is guaranteed to converge to an optimal policy $\bar{\pi}$ in any given Cheap Talk MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{M}, f, \mathcal{J}, \gamma \rangle$, since a Cheap Talk MDP is itself an MDP with an augmented state space $\mathcal{S} \times \mathcal{M}$ and augmented transition/reward functions that are defined to be independent from $\mathcal{M}$. Now $\bar{\pi}$ naturally induces a policy π on the no-channel MDP, given by $\pi(\cdot \mid s) \coloneqq \bar{\pi}(\cdot \mid s, f(s))$, and in particular $Q(s, a) = \bar{Q}(s, f(s), a)$ by independence of transitions and rewards from $\mathcal{M}$. Optimality of π follows directly from the Bellman equation

$$\begin{aligned}
Q(s, a) = \bar{Q}(s, f(s), a) &= \mathbb{E}_{s' \sim \mathcal{P}(\cdot \mid s, a), r \sim \mathcal{R}(\cdot \mid s, a)}\left[r + \gamma \max_{a' \in \mathcal{A}} \bar{Q}(s', f(s'), a')\right] \\
&= \mathbb{E}_{s' \sim \mathcal{P}(\cdot \mid s, a), r \sim \mathcal{R}(\cdot \mid s, a)}\left[r + \gamma \max_{a' \in \mathcal{A}} Q(s', a')\right]\,.
\end{aligned}$$

Now trajectories $\bar{\tau} = (s^k, f(s^k), a^k)_k$ and $\tau = (s^k, a^k)_k$ have identical probability and return under π and $\bar{\pi}$ respectively, so the Victim has expected return

$$\mathbb{E}_{\bar{\tau} \sim \bar{\pi}}\left[\mathcal{R}(\bar{\tau})\right] = \mathbb{E}_{\tau \sim \pi}\left[\mathcal{R}(\tau)\right]$$

which is the optimal expected return of the original no-channel MDP. □

A.3 FURTHER INFORMAL DISCUSSION

Consider a Cheap Talk MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \mathcal{M}, f, \mathcal{J} \rangle$. For a fixed training / testing run of the Victim on the MDP, the Adversary outputs a message $f(s)$ at each step according to a fixed deterministic function $f : \mathcal{S} \to \mathcal{M}$. Below we elaborate informally on the claims that Adversaries cannot (1) occlude the ground truth, (2) influence the environment dynamics / reward functions, (3) see the Victim's actions or parameters, (4) inject stochasticity, or (5) introduce non-stationarity.

(1) The message is *appended* to the state s and the Victim acts with full visibility of the ground truth (state) s according to its policy: $a \sim \pi(\cdot \mid s, f(s))$.

(2) The transition and reward functions $\mathcal{P}, \mathcal{R}$ are defined to be independent from $\mathcal{M}$. Formally we have $\mathcal{P}(\cdot \mid s, m, a) = \mathcal{P}(\cdot \mid s, m', a)$ for all $m, m' \in \mathcal{M}$ (similarly for $\mathcal{R}$), so the Adversary's choice of message $m = f(s)$ cannot influence $\mathcal{P}$ or $\mathcal{R}$.

(3) $f : \mathcal{S} \to \mathcal{M}$ is defined as a function of $\mathcal{S}$ only, so the Adversary cannot condition its policy based on the Victim's actions or parameters (i.e. it cannot see or influence them).

(4) f is a deterministic function, so $\pi(\cdot \mid s, f(s))$ is a distribution only on actions $\mathcal{A}$. The transition and reward functions are independent from f, so they are distributions only on state-action pairs $\mathcal{S} \times \mathcal{A}$. It follows that the Adversary injects no further stochasticity into the MDP.

(5) f is static for a fixed training / testing run, so $s_t = s_{t'}$ implies $f(s_t) = f(s_{t'})$ for all timesteps t, t' in the run. It follows that any given Victim policy π is stationary, namely $\pi(\cdot \mid s_t, f(s_t)) = \pi(\cdot \mid s_{t'}, f(s_{t'}))$ for all $s_t = s_{t'}$. Since $\mathcal{P}$ and $\mathcal{R}$ are stationary (as defined by a standard MDP) and independent from $\mathcal{M}$, their stationarity is also preserved.

Finally, we discuss the possibility of further weakening components of a Cheap Talk MDP, and conclude that all such variants (A-E) bring no advantage or reduce to regular MDPs.

(A) Removing the channel $\mathcal{M}$ or the policy $f : \mathcal{S} \to \mathcal{M}$ would result in the Victim being completely independent from the Adversary, so no adversarial influence could be exerted whatsoever.

(B) Restricting the capacity of $\mathcal{M}$ to a certain number of bits would further restrict an Adversary's range of influence, so one could say that the *truly* minimum-viable setting is to impose a set of size $|\mathcal{M}| = 1$. However, cheap talk is still cheap talk when varying capacity, and there is no reason to arbitrarily restrict the size to 1 if we are to apply our setting to complex environments likely requiring more than a single bit of communication to witness interesting results.

(C) Not allowing Adversaries to see states, namely removing $\mathcal{S}$ as inputs to f, yields a function $f : \{0\} \to \mathcal{M}$ which always outputs the same message $f(0) = m \in \mathcal{M}$. This is equivalent to the previous restriction of imposing a set $\mathcal{M}$ of size 1, since in this case any function $f : \mathcal{S} \to \mathcal{M}$ would have to output the unique element $f(s) = m$ for all input states s.

(D) The Adversary must have some objective function $\mathcal{J}$ in order for an adversarial setting to make sense – removing it would remove the Adversary's reason to exist, since it would have no incentive to learn parameters that influence the Victim according to some goal.

(E) Restricting the function class of objectives $\mathcal{J}$ is a valid minimisation of the setting, but simply restricts our interest in the setting itself. The setting should at the very least allow for adversarial objectives of the form $\mathcal{J} = \pm J$, as we consider in the train-time setting. In test-time, our aim is to show how Adversaries can exert arbitrary control over Victims despite cheap talk restrictions, and we therefore consider more general objective functions.

B PSEUDOCODE

Algorithm 2 Test-time ACT

Initialize train-time ACT parameters ϕ
Initialize test-time ACT parameters ψ
for $m = 0$ **to** M **do**
  Sample $\phi_n \sim \phi + \sigma\epsilon_n$ where $\epsilon_1, ..., \epsilon_N \sim \mathcal{N}(0, I)$
  Sample $\psi_n \sim \psi + \sigma\epsilon_n$ where $\epsilon_1, ..., \epsilon_N \sim \mathcal{N}(0, I)$
  for $n = 0$ **to** N **do**
    Initialize policy params θ
    rewards = []
    for $e = 0$ **to** E **do**
      s = env.reset()
      while not done **do**
        $m = f_{\phi_n}(s)$
        $\bar{s}$ = [s, m]
        $a \sim \pi_\theta(\cdot \mid \bar{s})$
        r, s = env.step(a)
      end while
      Update θ using PPO to maximise J
    end for
    for $i = 0$ **to** I **do**
      s = env.reset()
      while not done **do**
        $m = f_{\psi_n}(s)$
        $\bar{s}$ = [s, m]
        $a \sim \pi_\theta(\cdot \mid \bar{s})$
        r, s, done = env.step(a)
        $r_t^S = R^S(s, a)$
        rewards.append(r_t^S)
      end while
    end for
  end for
  Update ϕ using ES to maximise $\mathcal{J}$
  Update ψ using ES to maximise $\mathcal{J}$
end for

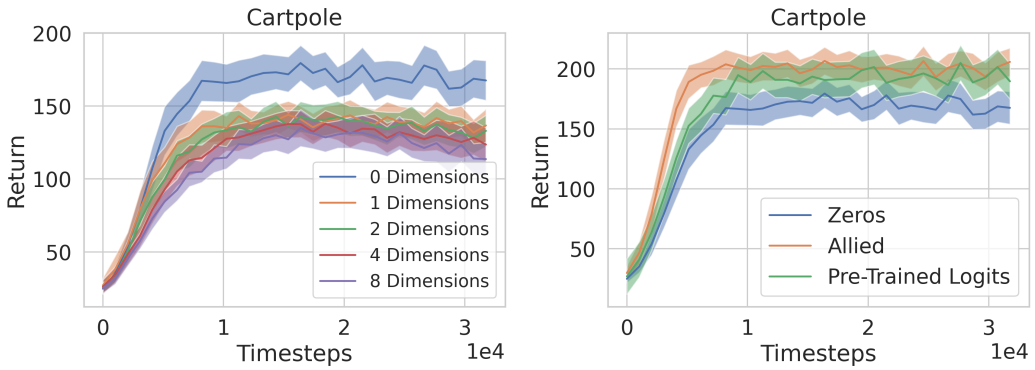
Algorithm 3 Test-time Oracle PPO ACT

Initialize train-time ACT parameters ϕ
Obtain trained ϕ, θ from Algorithm 2
Initialize test-time ACT parameters ψ^*
for $i = 0$ **to** I **do**
    s = env.reset()
    while not done **do**
        $m \sim \pi_{\psi^*}(\cdot \mid s)$
        $\bar{s}$ = [s, m]
        $a \sim \pi_\theta(\cdot \mid \bar{s})$
        r, s, done = env.step(a)
        $r_t^S = R^S(s, a)$
        rewards.append(r_t^S)
    end while
    Update ψ^* using PPO to maximise $\mathcal{J}$
end for

Algorithm 4 Test-time Random Shaper

Initialize train-time ACT parameters ϕ_{random}
Initialize policy params θ
rewards = []
for $e = 0$ **to** E **do**
    s = env.reset()
    while not done **do**
        $m = f_{\phi_{\text{random}}}(s)$
        $\bar{s}$ = [s, m]
        $a \sim \pi_\theta(\cdot \mid \bar{s})$
        r, s = env.step(a)
    end while
    Update θ using PPO to maximise J
end for
Initialize test-time ACT parameters ψ^*
for $i = 0$ **to** I **do**
    s = env.reset()
    while not done **do**
        $m \sim \pi_{\psi^*}(\cdot \mid s)$
        $\bar{s}$ = [s, m]
        $a \sim \pi_\theta(\cdot \mid \bar{s})$
        r, s = env.step(a)
        $r_t^S = R^S(s, a)$
        rewards.append(r_t^S)
    end while
    Update ψ^* using PPO to maximise $\mathcal{J}$
end for

C ABLATIONS



(a) (b)

Figure 7: (a) Ablations on the different number of cheap talk dimensions for the Adversary in Cartpole. We find that for a low-dimensional environment like Cartpole, the Adversary does not achieve much marginal improvement from increasing the number of channels, suggesting that there may be some limit to the amount that it can harm performance. (b) Comparing the ally with an Adversary that outputs pre-trained logits in Cartpole. We find that the allied ACT still performs better, implying that it is outputting features that are more useful than logits from a pre-trained policy. Error bars denote the standard error across 10 seeds of a Victim trained against a single meta-trained Adversary.

D PENDULUM ABLATION

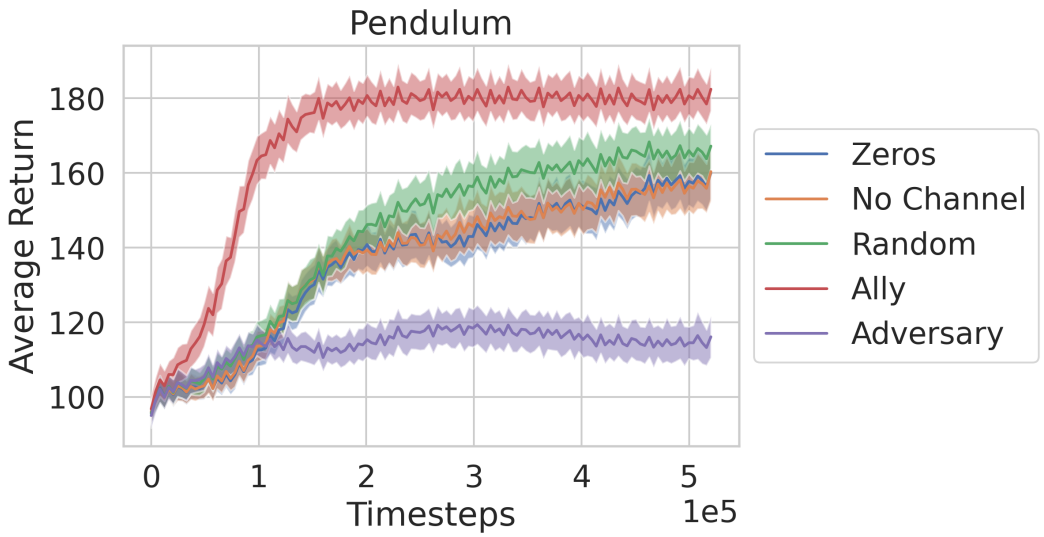


Figure 8: Interestingly, it seems like random network features improved performance in Pendulum. To make sure this was not due to network initialisation effects, we ran an ablation where we removed the cheap talk channel. It achieves about the same performance as a channel with zeros, which implies that the performance difference is not due to network initialisation.

E HYPERPARAMETER DETAILS

We report the hyperparameter values used for each environment in our experiments.

Table 1: Important parameters for the Cartpole environment

Parameter	Value
State Size	4
message Size	2
Number of Environments	4
Maximum Grad Norm	0.5
Number of Updates	32
Update Period	256
Outer Discount Factor γ	0.99
Number of Epochs per Update	16
PPO Clipping ϵ	0.2
General Advantage Estimation λ	0.95
Critic Coefficient	0.5
Entropy Coefficient	0.01
Learning Rate	0.005
Population Size	1024
Number of Generations	2049
Outer Agent (OA) Hidden Layers	2
OA Size of Hidden Layers	64
OA Hidden Activation Function	ReLU
OA Output Activation Function	Tanh
Inner Agent (IA) Actor Hidden Layers	2
IA Size of Actor Hidden Layers	32
IA Number of Critic Hidden Layers	2
IA Size of Critic Hidden Layers	32
IA Activation Function	Tanh
Number of Rollouts	4

Table 2: Important parameters for the Pendulum environment

Parameter	Value
State Size	3
message Size	2
Number of Environments	16
Maximum Grad Norm	0.5
Number of Updates	128
Update Period	256
Outer Discount Factor γ	0.95
Number of Epochs per Update	16
PPO Clipping ϵ	0.2
General Advantage Estimation λ	0.95
Critic Coefficient	0.5
Entropy Coefficient	0.005
Learning Rate	0.02
Population Size	768
Number of Generations	2049
Outer Agent (OA) Hidden Layers	2
OA Size of Hidden Layers	64
OA Hidden Activation Function	ReLU
OA Output Activation Function	Tanh
Inner Agent (IA) Actor Hidden Layers	1
IA Size of Actor Hidden Layers	32
IA Number of Critic Hidden Layers	1
IA Size of Critic Hidden Layers	32
IA Activation Function	Tanh
Number of Rollouts	4

Table 3: Important parameters for the Reacher environment

Parameter	Value
State Size	10
message Size	4
Number of Environments	32
Maximum Grad Norm	0.5
Number of Updates	256
Update Period	128
Outer Discount Factor γ	0.99
Number of Epochs per Update	10
PPO Clipping ϵ	0.2
General Advantage Estimation λ	0.95
Critic Coefficient	0.5
Entropy Coefficient	0.0005
Learning Rate	0.004
Population Size	128
Number of Generations	2049
Outer Agent (OA) Hidden Layers	2
OA Size of Hidden Layers	64
OA Hidden Activation Function	ReLU
OA Output Activation Function	Tanh
Inner Agent (IA) Actor Hidden Layers	2
IA Size of Actor Hidden Layers	128
IA Number of Critic Hidden Layers	2
IA Size of Critic Hidden Layers	128
IA Activation Function	ReLU
Number of Rollouts	4

Table 4: Important parameters for the Minatar environments

Parameter	Value
State Size	400
message Size	32
Number of Environments	64
Maximum Grad Norm	0.5
Number of Updates	1024
Update Period	256
Outer Discount Factor γ	0.99
Number of Epochs per Update	32
PPO Clipping ϵ	0.2
General Advantage Estimation λ	0.95
Critic Coefficient	0.5
Entropy Coefficient	0.01
Learning Rate	3e-4
Population Size	128
Number of Generations	256
Outer Agent (OA) Hidden Layers	2
OA Size of Hidden Layers	64
OA Hidden Activation Function	ReLU
OA Output Activation Function	Tanh
Inner Agent (IA) Actor Hidden Layers	2
IA Size of Actor Hidden Layers	256
IA Number of Critic Hidden Layers	2
IA Size of Critic Hidden Layers	256
IA Activation Function	ReLU
Number of Rollouts	1