Theorem 2: Under free communication, consider a team of agents using a coordination policy: \( \pi_{\Sigma}(b^t_{\Sigma}) \equiv \Omega_t^t \). If the domain-level policy \( \pi_A \) maximizes \( V^T(\pi_A, \pi_{\Sigma}) \), then this combined policy is dominant over any other policies. In other words, for all policies, \( \pi'_A, \pi'_{\Sigma} \), \( V^T(\pi'_A, \pi'_{\Sigma}) \geq V^T(\pi_A, \pi_{\Sigma}) \).

Proof: Suppose we have some other coordination policy, \( \pi'_{\Sigma} \), that specifies something other than complete communication (e.g., keeping quiet, lying). Suppose that there is some domain-level policy, \( \pi'_A \), that allows the team to attain some expected reward, \( K \), when used in combination with \( \pi'_{\Sigma} \). Then, we can construct a domain-level policy, \( \pi_A \), such that the team attains the same expected reward, \( K \), when used in conjunction with the complete communication policy, \( \pi_{\Sigma} \), as defined in the statement of Theorem 2.

The coordination policy, \( \pi'_{\Sigma} \), produces a different set of belief states (denoted \( b^t_{\Sigma} \) and \( b^t_{\Sigma,i} \)) than those for \( \pi_{\Sigma} \) (denoted \( b^t_{\Sigma} \) and \( b^t_{\Sigma,i} \)). In particular, we use state estimator functions, \( SE_{\Sigma,i}^t \) and \( SE_{\Sigma}^t \), as defined in Equations 2 and 3, to generate \( b^t_{\Sigma} \) and \( b^t_{\Sigma,i} \). Each belief state is a complete history of observation and communication pairs for each agent. On the other hand, under the complete communication of \( \pi_{\Sigma} \), the post-communication state estimator function reduces to:

\[
SE_{\Sigma,i}(\langle \Omega^0, \ldots, \Omega^{t-1}, \Omega_t^t \rangle, \Sigma^t) = \langle \Omega^0, \ldots, \Omega^{t-1}, \Sigma^t \rangle
\]

Since each agent’s message is exactly its observation,

\[
= \langle \Omega^0, \ldots, \Omega^{t-1}, \Sigma^t \rangle
\]

Thus, \( \pi_A \) is defined over a different set of belief states than \( \pi'_A \). In order to determine an equivalent \( \pi_A \), we must first define a recursive mapping, \( m_i \), that translates the belief states defined by \( \pi_{\Sigma} \) into those defined by \( \pi'_{\Sigma} \):

\[
m_i(b^t_{\Sigma,i})
\]

The belief state at time \( t \) is a sequence of observations, which we can divide into the observations before time \( t \) and the observation at time \( t \). The observations before time \( t \) correspond exactly to the belief state at time \( t - 1 \).

\[
m_i(b^t_{\Sigma,i} \cdot \Omega^t)
\]

The combined observation at time \( t \) includes agent \( i \)’s observation, as well as everyone else’s observations.

\[
m_i(b^t_{\Sigma,i} \cdot \Omega^t)
\]

We can “distribute” the mapping function over the two components in the tuple. Under \( \pi'_{\Sigma} \), the agents would not communicate their observations, but instead some other set of messages.

\[
m_i(b^t_{\Sigma,i} \cdot \Omega^t)
\]

We can break these messages down across the individual agents.

\[
m_i(b^t_{\Sigma,i} \cdot \Omega^t)
\]

Each agent, \( j \), selects its message based on following the communication policy, \( \pi'_j \), from its pre-communication belief state.

\[
m_i(b^t_{\Sigma,i} \cdot \Omega^t)
\]
We can generate each agent’s pre-communication belief state by applying the pre-communication state-estimator function to each agent’s previous post-communication belief state and its most recent observation (which we know from its previous message, under full communication).

\[ m_i(b_{t-1}^{\bullet}) \left\{ \Omega_i, \prod_{j \in a} \pi_j^{t} \bigl( \mathcal{SE}_{j}^{t} \bigl( b_{j \in \alpha}^{t-1}, \Omega_j^{t} \bigr) \bigl) \right\} \]

Finally, we can compute each agent’s previous post-communication belief state by again applying the mapping function.

\[ m_i(b_{t-1}^{\bullet}) \left\{ \Omega_i, \prod_{j \in a} \pi_j^{t} \bigl( \mathcal{SE}_{j}^{t} \bigl( m_j(b_{j \in \alpha}^{t-1}), \Omega_j^{t} \bigr) \bigl) \right\} \quad (2) \]

Given this mapping, we then specify: \( \pi_A(b_{t}^{\bullet}) = \pi_A(m_i(b_{t}^{\bullet})) \). Executing this domain-level policy, in conjunction with the coordination policy, \( \pi_\Sigma \), results in the identical behavior as execution of the alternate policies, \( \pi_A' \) and \( \pi_\Sigma' \). Therefore, the team following the policies, \( \pi_A \) and \( \pi_\Sigma \), will achieve the same expected value of \( K \), as under \( \pi_A' \) and \( \pi_\Sigma' \). \( \square \)