Appendix

Proof of Lemma 1

Proof. Consider the messaging scheme $\pi^*(m_a|s)$ defined above, which is clearly Markov since it does not depend on any history information. We perturb the scheme a bit and parameterize the perturbation by $\epsilon > 0$. We show that if ϵ is small enough, the perturbed scheme is persuasive.

We construct a perturbed messaging scheme π as follows. We leave the scheme untouched for any state $s \in S \setminus \{s_{i_1}\}$, i.e.,

$$\pi(m_a|s) = \begin{cases} 1 & \text{if } a = \beta_r^*(s) \\ 0 & \text{otherwise} \end{cases}.$$

And for s_{i_1}, we set

$$\pi(m_a|s_{i_1}) = \begin{cases} 1-\epsilon & \text{if } a = a_{i_1} \\ \epsilon & \text{if } a = a_{i_2} \\ 0 & \text{otherwise} \end{cases}.$$

The perturbed scheme is also Markov. For Markov schemes, the persuasiveness constraint (5) can be reduced to the following:

$$\begin{aligned} &\sum_{s\in S} \rho_h(s)\pi(m_a|s)u_r(s,a) \\ \geq &\sum_{s\in S} \rho_h(s)\pi(m_a|s)u_r(s,a'), \forall a, a' \in A, \forall h. \end{aligned} \tag{11}$$

Thus the original scheme $\pi^*(m_a|s)$ satisfies:

$$\sum_{s\in S_{i_1}} \rho_h(s)\left[u_r(s,a_{i_1}) - u_r(s,a')\right] > 0, \forall a' \neq a_{i_1}, \forall h, \tag{12}$$

where we define $S_j = \{s \mid \beta_r^*(s) = a_j\}$. Note that we change the weak inequality in Equation (11) to the strict one here because we have $u_r(s_{i_1}, a_{i_1}) > u_r(s_{i_1}, a'), \forall a' \in A$ according to our assumption. Similarly, we have:

$$\sum_{s\in S_{i_2}} \rho_h(s)\left[u_r(s,a_{i_2}) - u_r(s,a')\right] > 0, \forall a' \neq a_{i_2}, \forall h. \tag{13}$$

Now we show that the perturbed scheme satisfies constraint (11) for a small enough ϵ. When the sender sends message $m_{a_{i_1}}$, the receiver knows, according to the definition of π, that the only possible states are those in S_{i_1}. Thus, to ensure persuasiveness, we need to guarantee that for any action a' and history h, the following holds:

$$\begin{aligned} &\sum_{s\in S_{i_1}\setminus\{s_{i_1}\}} \rho_h(s)\left[u_r(s,a_{i_1}) - u_r(s,a')\right] \\ &\quad + \rho_h(s_{i_1})(1-\epsilon)\left[u_r(s_{i_1},a_{i_1}) - u_r(s_{i_1},a')\right] \\ =&\sum_{s\in S_{i_1}} \rho_h(s)\left[u_r(s,a_{i_1}) - u_r(s,a')\right] \\ &\quad - \rho_h(s_{i_1})\epsilon\left[u_r(s_{i_1},a_{i_1}) - u_r(s_{i_1},a')\right] \\ \geq& 0. \end{aligned}$$

This can be done by setting

$$0 < \epsilon \leq \min_{a',h} \left\{ \frac{\sum_{s\in S_{i_1}} \rho_h(s)\left[u_r(s,a_{i_1}) - u_r(s,a')\right]}{\rho_h(s_{i_1})\left[u_r(s_{i_1},a_{i_1}) - u_r(s_{i_1},a')\right]} \right\}, \tag{14}$$

which is well-defined since $u_r(s_{i_1}, a_{i_1}) > u_r(s_{i_1}, a'), \forall a' \neq a_{i_1}$. And the right-hand side is strictly positive according to Equation (12).

When the sender sends $m_{a_{i_2}}$, the set of possible states is $S_{i_2} \cup \{s_{i_1}\}$. Thus the persuasiveness constraint in this case becomes:

$$\begin{aligned} &\rho_h(s_{i_1})\epsilon\left[u_r(s_{i_1},a_{i_2}) - u_r(s_{i_1},a')\right] \\ &\quad + \sum_{s\in S_{i_2}} \rho_h(s)\left[u_r(s,a_{i_2}) - u_r(s,a')\right] \geq 0, \forall a', \forall h. \end{aligned}$$

That the second term is strictly positive according to Equation (13), while the first term can be negative since a_{i_1} is the unique maximizer of $u_r(s_{i_1}, a)$, i.e., $u_r(s_{i_1}, a_{i_2}) < u_r(s_{i_1}, a_{i_1})$. For any a' with $u_r(s_{i_1}, a_{i_2}) \geq u_r(s_{i_1}, a')$, setting any positive ϵ will do. But for a' with $u_r(s_{i_1}, a_{i_2}) < u_r(s_{i_1}, a')$, we need to make ϵ small enough to ensure the above inequality. Thus we can set:

$$0 < \epsilon \leq \left| \min_{a',h} \left\{ \frac{\sum_{s\in S_{i_2}} \rho_h(s)\left[u_r(s,a_{i_2}) - u_r(s,a')\right]}{\rho_h(s_{i_1})\left[u_r(s_{i_1},a_{i_2}) - u_r(s_{i_1},a')\right]} \right\} \right|. \tag{15}$$

Note that the term inside the absolute value function is strictly negative.

When the sender sends messages other than $m_{a_{i_1}}$ and $m_{a_{i_2}}$, the persuasiveness constraints are the same as those of the original scheme, and thus already satisfied. Therefore, to guarantee persuasiveness, we can choose any ϵ that satisfies both Equation (14) and (15). And According to our analysis, there are clearly infinitely many such choices. ☐

Proof of Theorem 1

Proof. Define a new scheme based on M_A as follows:

$$\pi'(m_a|h,s) = \sum_{m\in M_a(h)} \pi(m|h,s).$$

This new scheme induces a new MDP for the receiver. We claim that the value function is

$$\begin{aligned} &V_2^{\pi'}(h, m_a) \\ =&\sum_{m\in M_a(h)} V_2^{\pi}(h,m) \frac{\sum_s \pi(m|h,s)\rho_h(s)}{\sum_{m'\in M_a(h)}\sum_s \pi(m'|h,s)\rho_h(s)}, \end{aligned} \tag{16}$$

and that the receiver's strategy $a = \beta'(h, m_a)$ is optimal, hence persuasive.

Denote by $h' = h + (s, a)$. To prove the claims, it suffices to show that the value function satisfies the Bellman equation:

$$
\begin{aligned}
&V_2^{\pi'}(h, m_a)\\
&= \arg\max_{\hat{a}} \Bigg\{ \sum_s \rho_h(s|m_a, h) \Bigg[u_r(s, \hat{a})\\
&\quad + \gamma \sum_{s'} P(s'|s, \hat{a}) \sum_{a'} \pi'(m_{a'}|h', s') V_2^{\pi'}(h', m_{a'}) \Bigg]\Bigg\},
\end{aligned} \tag{17}
$$

and that using $a = \beta'(h, m_a)$ maximizes the right-hand side of the above equation.

Since $\beta(h, m)$ is the receiver's optimal strategy in the original MDP, we have that

$$
\begin{aligned}
&V_2^{\pi}(h, m)\\
&= \arg\max_{\hat{a}} \Bigg\{ \sum_s \rho_h(s|m, h) \Bigg[u_r(s, \hat{a})\\
&\quad + \gamma \sum_{s'} P(s'|s, \hat{a}) \sum_{a'} \pi'(m|h', s') V_2^{\pi}(h', m) \Bigg]\Bigg\}.
\end{aligned} \tag{18}
$$

And for any $m \in M_a(h)$, by definition, action a maximizes the right-hand side.

Combining Equation (18) and Equation (16) gives:

$$
\begin{aligned}
&V_2^{\pi'}(h, m_a)\\
&= \sum_{m \in M_a(h)} \frac{\left[\sum_s \pi(m|h, s)\rho_h(s)\right]\left[\sum_s \rho_h(s|m, h) u_r(s, a)\right]}{\sum_{m' \in M_a(h)} \sum_s \pi(m'|h, s)\rho_h(s)}\\
&\quad + \sum_{m \in M_a(h)} \frac{\sum_s \pi(m|h, s)\rho_h(s)}{\sum_{m' \in M_a(h)} \sum_s \pi(m'|h, s)\rho_h(s)} \gamma \sum_s \Bigg\{\\
&\rho_h(s|h, m) \sum_{s'} P(s'|s, a) \left[\sum_{m'} \pi(m'|h', s') V_2^{\pi}(h', m')\right] \Bigg\}
\end{aligned} \tag{19}
$$

Consider the first term:

$$
\begin{aligned}
&\sum_{m \in M_a(h)} \frac{\left[\sum_s \pi(m|h, s)\rho_h(s)\right]\left[\sum_s \rho_h(s|m, h) u_r(s, a)\right]}{\sum_{m' \in M_a(h)} \sum_s \pi(m'|h, s)\rho_h(s)}\\
&= \sum_{m \in M_a(h)} \frac{\sum_s \rho_h(s)\pi(m|h, s) u_r(s, a)}{\sum_{m' \in M_a(h)} \sum_s \pi(m'|h, s)\rho_h(s)}\\
&= \frac{\sum_s \rho_h(s)\pi(m_a|h, s) u_r(s, a)}{\sum_s \pi(m_a|h, s)\rho_h(s)}\\
&= \sum_s \rho_h(s|h, m_a) u_r(s, a),
\end{aligned} \tag{20}
$$

The second equation is obtained by plugging in Equation (7), and in the last equation,

$$
\begin{aligned}
\rho_h(s|h, m_a) &= \frac{\rho_h(s)\pi'(m_a|h, s)}{\sum_{s'} \rho_h(s')\pi'(m_a|h, s')}\\
&= \frac{\rho_h(s)\sum_{m \in M_a(h)} \pi(m|h, s)}{\sum_{s'} \rho_h(s')\sum_{m \in M_a(h)} \pi(m|h, s)}
\end{aligned}
$$

is the receiver's posterior belief in the new MDP. Now consider the second term of Equation (19). Define:

$$
V(h') = \sum_{s'} P(s'|s, a) \left[\sum_{m'} \pi(m'|h', s') V_2^{\pi}(h', m')\right].
$$

Note that the state transition $P(s'|s, a)$ is equivalent to $\rho_{h'}(s')$. According to Equation (16), we have:

$$
\begin{aligned}
&V_2^{\pi'}(h', m_a) \sum_{m' \in M_a(h')} \sum_{s'} \pi(m'|h', s')\rho_{h'}(s')\\
&= \sum_{m \in M_a(h')} V_2^{\pi}(h', m) \sum_{s'} \pi(m|h', s')\rho_{h'}(s').
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
V(h') &= \sum_{s'} \rho_{h'}(s') \sum_m V_2^{\pi}(h', m)\pi(m|h', s')\\
&= \sum_m V_2^{\pi}(h', m) \sum_{s'} \pi(m|h', s')\rho_{h'}(s')\\
&= \sum_{a'} \sum_{m \in M_{a'}(h')} V_2^{\pi}(h', m) \sum_{s'} \pi(m|h', s')\rho_{h'}(s')\\
&= \sum_{a'} V_2^{\pi'}(h', m_{a'}) \sum_{s'} \sum_{m' \in M_a(h')} \pi(m'|h', s')\rho_{h'}(s')\\
&= \sum_{s'} \rho_{h'}(s') \sum_{a'} \pi'(m_{a'}|h', s') V_2^{\pi'}(h', m_{a'}).
\end{aligned}
$$

Thus the second term of Equation (19) can be written as:

$$
\gamma \sum_{m \in M_a(h)} \frac{\sum_s \pi(m|h, s)\rho_h(s)}{\sum_s \pi'(m_a|h, s)\rho_h(s)} \sum_s \rho_h(s|h, m) V(h').
$$

Note that $\sum_s \pi'(m_a|h, s)\rho_h(s)$ does not depend on s. Using Equation (7), we have:

$$
\begin{aligned}
&\gamma \sum_{m \in M_a(h)} \sum_s \frac{\rho_h(s)\pi(m|h, s)}{\sum_s \pi'(m_a|h, s)\rho_h(s)} V(h')\\
&= \gamma \sum_s \frac{\rho_h(s)\pi'(m_a|h, s)}{\sum_s \pi'(m_a|h, s)\rho_h(s)} V(h')\\
&= \gamma \sum_s \rho_h(s|h, m_a) V(h').
\end{aligned}
$$

Put both terms back to Equation (19), we get:

$$
\begin{aligned}
&V_2^{\pi'}(h, m_a)\\
&= \sum_s \rho_h(s|h, m_a) u_r(s, a) + \gamma \sum_s \rho_h(s|h, m_a) V(h')\\
&= \sum_s \rho_h(s|h, m_a) u_r(s, a) + \gamma \sum_s \rho_h(s|h, m_a) \sum_{s'} \Bigg\{\\
&\rho_{h'}(s') \sum_{a'} \pi'(m_{a'}|h', s') V_2^{\pi'}(h', m_{a'}) \Bigg\}\\
&= \sum_s \rho_h(s|h, m_a) \Bigg\{ u_r(s, a) + \gamma \sum_s \rho_h(s|h, m_a) \sum_{s'} \Bigg\{\\
&\rho_{h'}(s') \sum_{a'} \pi'(m_{a'}|h', s') V_2^{\pi'}(h', m_{a'}) \Bigg\}.
\end{aligned}
$$

Note that the above equation depends crucially on Equation (18). And in the new MDP, for any message $m \in M_a(h)$, choosing action a maximizes the right-hand side of Equation (18). This means in the right-hand side of the above equation, action a is also the best choice. Therefore, We have Equation (17). $\square$

Additional Experiment results

We report the experimental results in a setting where the sender can use threat-based strategies. We re-use the game instances generated for experiments with the standard, non-threat-based k-memory strategies.

Running time. The running time of the bi-linear program method is listed in Table 5 and Table 6. As shown in Table 5, Gurobi gives feasible solutions for all game instances of size 2 but failed for almost all games with a larger size. Compared with Table 1, this implies that finding a threat-based strategy is much more difficult for Gurobi. This also aligns with our intuitions as the strategy space is larger in the threat-based setting (see Section 7).

Table 6 shows similar patterns as Table 2: the number of solvable games decreases as the memory length k increases. Again, Gurobi finds much fewer threat-based solutions than non-threat-based ones due to the larger search space.

The results for our algorithm are shown in Table 7 and Table 8. Our algorithm gives feasible solutions for all game instances within 30 minutes. In fact, it takes about only 30 seconds for our algorithm to output a feasible solution for most instances. Compared with Table 3 and 4, our algorithm takes only a slightly longer time to find a feasible threat-based solution.

We apply our algorithm to larger games and report the results in Figure 7. Our algorithm can solve games with sizes up to 12×12 within 30 minutes, which is similar to the case of non-threat-based strategies as shown in Figure 4. Comparing Figure 4 and 7, we can see that the average utility of threat-based strategies is larger than non-threat-based ones. This is also because the strategy space is larger with threat-based strategies.

Performance. Figure 6 compares the performance of both algorithms in 2×2 games with different discount factors and memory lengths. Our algorithm achieves almost identical performance compared to the bi-linear program method. Since our results are averaged over the instances that are solved by both algorithms, one reason for the identi-

Table 5: Number of games that Gurobi gives a feasible solution within 30 mins for $k = 1$.

		Game size					
		2	3	4	5	6	8
	0.9	20	2	0	0	0	0
	0.7	20	0	0	0	0	0
γ	0.5	20	3	0	0	0	0
	0.3	20	6	0	0	0	0
	0.1	20	2	1	0	0	0

Table 6: Number of games that Gurobi gives a feasible solution within 30 mins for game size 2×2.

		Memory length k					
		1	2	3	4	5	6
	0.9	20	8	8	8	5	1
	0.7	20	11	8	6	5	3
γ	0.5	20	11	8	7	5	3
	0.3	20	12	9	8	4	3
	0.1	20	16	10	9	5	2

Table 7: Average running time (in seconds) of our threat-based algorithm for $k = 1$.

		Game size			
		2	3	4	5
	0.9	0.619	2.901	10.051	28.017
	0.7	0.625	2.963	10.278	28.731
γ	0.5	0.620	2.929	10.130	28.219
	0.3	0.621	2.930	10.174	28.232
	0.1	0.640	3.009	10.439	29.118

Table 8: Average running time (in seconds) of our threat-based algorithm for game size 2×2.

		Memory length k			
		1	2	3	4
	0.9	0.631	2.469	9.893	39.054
	0.7	0.630	2.482	9.918	38.935
γ	0.5	0.634	2.479	9.847	38.894
	0.3	0.628	2.471	9.782	38.636
	0.1	0.638	2.514	9.979	39.071

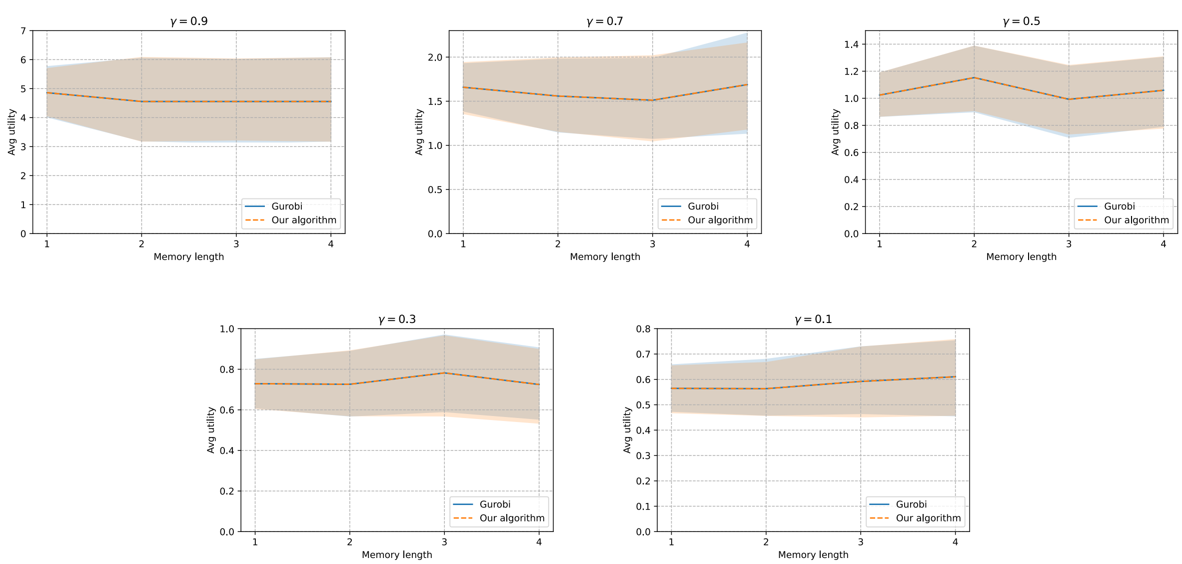

Figure 6: Average sender utility obtained by different threat-based algorithms in 2×2 games.

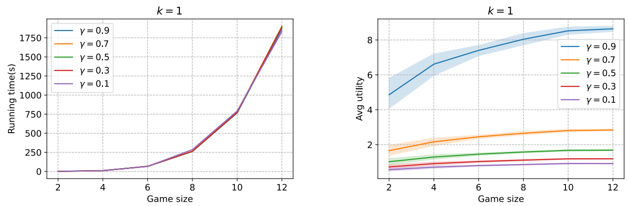

Figure 7: Average running time and utility of our threat-based algorithm for $k = 1$ in games with different sizes.

cal performance is that in this threat-based setting, Gurobi is only able to solve much fewer game instances (see Table 6 and 4 for details). Similar to Figure 3, in general, increasing the memory length does not lead to a higher expected utility, i.e., using a more complicated strategy may not benefit the sender too much. Also note that, since both algorithms are only able to give feasible solutions, a longer memory length may sometime result in a lower utility in our experiments.

Table 9: Average sender utility obtained by different threat-based algorithms with memory length $k = 1$, where - denotes that there is no game instance can be solved in 30 minutes.

γ	Algorithm	Game size			
		2	3	4	5
0.9	Our algorithm	4.859	8.232	6.613	7.158
	Gurobi	4.859	7.994	-	-
0.7	Our algorithm	1.658	2.204	2.163	2.395
	Gurobi	1.658	-	-	-
0.5	Our algorithm	1.023	1.587	1.291	1.440
	Gurobi	1.023	1.445	-	-
0.3	Our algorithm	0.728	1.065	0.914	1.039
	Gurobi	0.728	1.061	-	-
0.1	Our algorithm	0.564	0.892	0.712	0.811
	Gurobi	0.564	0.892	-	-

The performance comparison for different game sizes is shown in Table 9. The "-" symbol indicates that no feasible solution is found for any of the 20 game instances. Our algorithm also achieves similar performance compared to the bilinear method in instances solved by both algorithms. Similar to Figure 2, the sender is able to obtain larger expected utilities in larger games in general.