RESEARCH ARTICLE

COMPUTER SCIENCE

Human-level play in the game of *Diplomacy* by combining language models with strategic reasoning

Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin[1]‡, Noam Brown[1]*‡, Emily Dinan[1]*‡, Gabriele Farina[1], Colin Flaherty[1]‡, Daniel Fried[1,2], Andrew Goff[1], Jonathan Gray[1]‡, Hengyuan Hu[1,3]‡, Athul Paul Jacob[1,4]‡, Mojtaba Komeili[1], Karthik Konath[1], Minae Kwon[1,3], Adam Lerer[1]*‡, Mike Lewis[1]*‡, Alexander H. Miller[1]‡, Sasha Mitts[1], Adithya Renduchintala[1]‡, Stephen Roller[1], Dirk Rowe[1], Weiyan Shi[1,5]‡, Joe Spisak[1], Alexander Wei[1,6], David Wu[1]‡, Hugh Zhang[1,7]‡, Markus Zijlstra[1]

Despite much progress in training artificial intelligence (AI) systems to imitate human language, building agents that use language to communicate intentionally with humans in interactive environments remains a major challenge. We introduce Cicero, the first AI agent to achieve human-level performance in *Diplomacy*, a strategy game involving both cooperation and competition that emphasizes natural language negotiation and tactical coordination between seven players. Cicero integrates a language model with planning and reinforcement learning algorithms by inferring players' beliefs and intentions from its conversations and generating dialogue in pursuit of its plans. Across 40 games of an anonymous online *Diplomacy* league, Cicero achieved more than double the average score of the human players and ranked in the top 10% of participants who played more than one game.

A major long-term goal for the field of artificial intelligence (AI) is to build agents that can plan, coordinate, and negotiate with humans in natural language. Although much progress has been made in language models that imitate human language (*1*), effective negotiation agents must go beyond this by understanding the beliefs, goals, and intentions of their partner; planning joint actions that account for their partner's goals; and persuasively and intentionally communicating these proposals.

We present Cicero, an AI agent that achieved human-level performance in the strategy game *Diplomacy*. In *Diplomacy*, seven players conduct private natural language negotiations to coordinate their actions to both cooperate and compete with each other. By contrast, prior major successes for multi-agent AI have been in purely adversarial environments such as chess (*2*), Go (*3*), and poker (*4*), in which communication has no value. For these reasons, *Diplomacy* has served as a challenging benchmark for multi-agent learning (*5*–*8*).

[1]Meta AI, 1 Hacker Way, Menlo Park, CA, USA. [2]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA. [3]Department of Computer Science, Stanford University, Stanford, CA, USA. [4]Computer Science and Artificial Intelligence Laboratory, Massachusetts Insititute of Technology, Cambridge, MA, USA. [5]Department of Computer Science, Columbia University, New York, NY, USA. [6]Department of Computer Science, University of California, Berkeley, Berkeley, CA, USA. [7]EconCS Group, Harvard University, Cambridge, MA, USA.
***Corresponding author. Email: noambrown@meta.com (N.B.); edinan@meta.com (E.D.); alerer@meta.com (A.L.); mikelewis@meta.com (M.L.)** †FAIR consists of all listed authors. There are no additional authors or collaborators. ‡These authors contributed equally to this work.

Cicero couples a controllable dialogue module with a strategic reasoning engine. At each point in the game, Cicero models how the other players are likely to act on the basis of the game state and their conversations. It then plans how the players can coordinate to their mutual benefit and maps these plans into natural language messages.

We entered Cicero anonymously in 40 games of *Diplomacy* in an online league of human players between 19 August and 13 October 2022. Over the course of 72 hours of play that involved sending 5277 messages, Cicero ranked in the top 10% of participants who played more than one game.

Challenges of human-AI cooperation in *Diplomacy*

Almost all prior AI breakthroughs in games have been in two-player zero-sum (2p0s) settings, including chess (*2*), Go (*3*), heads-up poker (*9*, *10*), and *StarCraft* (*11*, *12*). In finite 2p0s games, certain reinforcement learning (RL) algorithms that learn by playing against themselves—a process known as self-play—will converge to a policy that is unbeatable in expectation in balanced games (*13*). In other words, any finite 2p0s game can be solved through self-play with sufficient compute and model capacity.

However, in games that involve cooperation, self-play without human data is no longer guaranteed to find a policy that performs well with humans, even with infinite compute and model capacity, because the self-play agent may converge to a policy that is incompatible with human norms and expectations. This effect can be clearly seen in settings that involve language, in which prior work found that self-play produced uninterpretable language despite achieving high task success for the agents (*14*, *15*). Even in dialogue-free versions of *Diplomacy*, we found that a self-play algorithm that achieved superhuman performance in 2p0s versions of the game performed poorly in games with multiple human players owing to learning a policy inconsistent with the norms and expectations of potential human allies (*16*, *17*). Thus, a major challenge in *Diplomacy* is to develop a way to harness the potential benefits of self-play in a way that leads to human-compatible language and behavior.

The challenge of maintaining human-interpretable communication is particularly acute in *Diplomacy*, in which our agent sent and received an average of 292 messages per game (fig. S8). Messages in the game often involve coordinating precise plans, and any miscommunication can result in their failure. Each message an agent sends must be grounded in (be contextually appropriate and consistent with) lengthy dialogue histories, game states—including proposed hypothetical states—and goals. If messages are inaccurately grounded, humans may ask the agent to explain its errors (a challenging task that may lead to further mistakes) or choose to cooperate with others instead. Further, repeated messaging creates feedback loops, in which the language model imitates the style of its own previous messages—for example, sending a short or incoherent message will increase the likelihood of such messages in the future (*18*). Past work on strategic dialogue systems has avoided these issues by focusing on simpler settings (*14*, *19*–*21*), which involve only a single human partner, shorter dialogue histories, and simpler strategies.

Last, *Diplomacy* is a particularly challenging domain because success requires building trust with others in an environment that encourages players to not trust anyone. Each turn's actions occur simultaneously after nonbinding, private negotiations. To succeed, an agent must account for the risk that players may not stay true to their word, or that other players may themselves doubt the honesty of the agent. For this reason, an ability to reason about the beliefs, goals, and intentions of others and an ability to persuade and build relationships through dialogue are powerful skills in *Diplomacy*.

The game of *Diplomacy*

Diplomacy is a board game in which seven players compete to control supply centers (SCs) on a map, by moving their units into them. A player wins by controlling a majority of SCs. The game may also end when all remaining players agree to a draw, or a turn limit is reached, in which case scores are determined by the number of SCs each player controls.

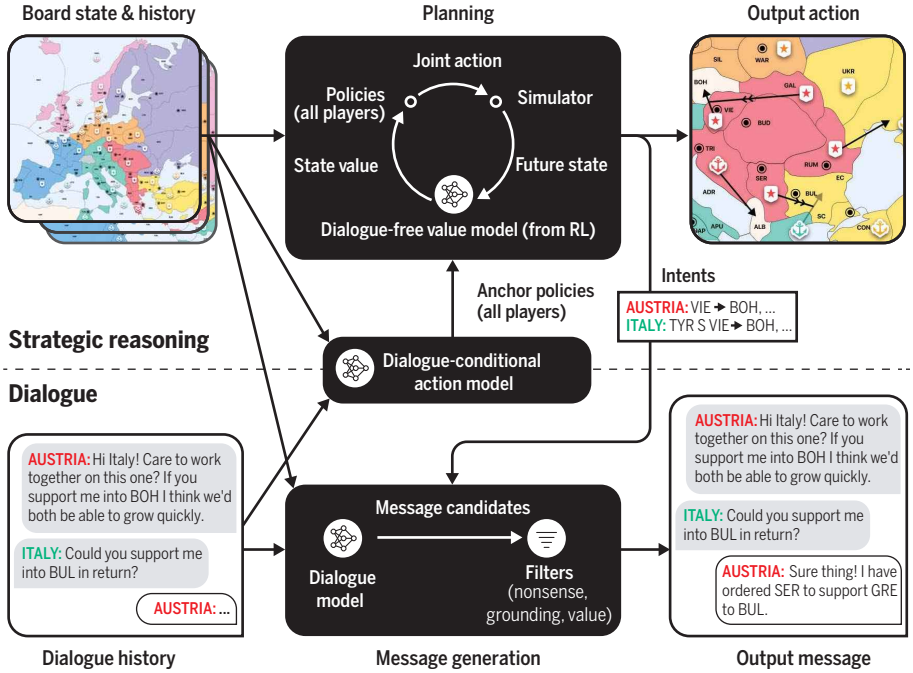


Fig. 1. Architecture of Cicero. Cicero predicts likely human actions for each player according to the board state and dialogue, using that as the starting point for a planning algorithm using RL-trained models. The output of planning is an action for the agent as well as beliefs about other players' actions, which are used to select intents for a dialogue model to condition on. Generated message candidates undergo several filtering steps before a final message is sent.

Each turn, all players engage in private pairwise free-form dialogue with the others during a negotiation period, and then all players simultaneously choose an action comprising one order per unit they control. A unit may support other units, including those of another player, which forms the basis for much of the negotiation in *Diplomacy*. A detailed description of the rules is provided in the supplementary materials (SM), materials and methods, section C.

Overview of Cicero

At a high level, Cicero combines a dialogue module with a strategic reasoning module, along with a filtering process to reject low-quality messages. A diagram of Cicero is provided in Fig. 1.

Dialogue

Cicero generates dialogue using a pretrained language model that was further trained on dialogue data from human games of *Diplomacy*. Crucially, in addition to being grounded in both the dialogue history and game state, the dialogue model was trained to be controllable through intents, which we here define to be a set of planned actions for the agent and its speaking partner. This was accomplished by automatically augmenting the human data with inferred intents and using this information as further conditioning during training. For example, intents showing the agent moving into the territory Bulgaria ("BUL") with support from its speaking partner might yield a message such as "Could you support me into BUL in return?" Grounding in intents relieved the dialogue model of most of the responsibility for learning which actions were legal and strategically beneficial. In particular, this control provided an interface between the dialogue generation and strategic reasoning.

Strategic reasoning

Cicero uses a strategic reasoning module to intelligently select intents and actions. This module runs a planning algorithm that predicts the policies of all other players on the basis of the game state and dialogue so far, accounting for both the strength of different actions and their likelihood in human games, and chooses an optimal action for Cicero that is based on those predictions. Planning relies on a value and policy function trained through self-play RL that penalized the agent for deviating too far from human behavior, to maintain a human-compatible policy. During each negotiation period, intents are recomputed every time Cicero sends or receives a message. At the end of each turn, Cicero plays its most recently computed intent.

Message filtering

Cicero passes each generated message through several filters designed to limit messages that are nonsensical, inconsistent with intents, or strategically poor.

Methods

Data

We obtained a dataset of 125,261 games of *Diplomacy* played online at webDiplomacy.net. Of these, 40,408 games contained dialogue, with a total of 12,901,662 messages exchanged between players. Player accounts were de-identified, and automated redaction of personally identifiable information (PII) was performed by webDiplomacy. We refer to this dataset hereafter as WebDiplomacy.

Intent-controlled dialogue

Cicero generates messages through a neural generative Diplomacy dialogue model that was trained to be controllable through a set of intents.

Imitation dialogue model

We took R2C2 (*22*) as our base model—a 2.7 billion–parameter Transformer-based (*23*) encoder-decoder model pretrained on text from the internet by using a BART denoising objective (*24*). The base pretrained model was then further trained on WebDiplomacy (Methods, Data) through standard maximum likelihood estimation. Specifically, with a dataset $\mathcal{D} = \{[\mathbf{x}^{(i)}, \mathbf{y}^{(i)}]\}$, the model was trained to predict a dialogue message $\mathbf{y}^{(i)}$ from player $\mathcal{A}$ to player $\mathcal{B}$ at time t, given all of the following represented as text $\mathbf{x}^{(i)}$: dialogue history (all messages exchanged between player $\mathcal{A}$ and the six other players up to time t); game state and action history (current game state and recent action history); player rating (rating for $\mathcal{A}$ corresponding to Elo rating computed from games in WebDiplomacy); game and message metadata (additional info about game settings and the current message, such as time since the last message, and current turn). Additionally, the model conditions on intents (a set of proposed actions for players $\mathcal{A}$ and $\mathcal{B}$ for the current turn and future turns, representing the intent for message $\mathbf{y}^{(i)}$). Further details on the training data, training procedure, relevant hyperparameters, sampling procedures, and other inference-time methods are provided in the SM, section D.1.

During play, we used additional modules governing when to speak and to whom, which are described in the SM, section D.4.

Controllable dialogue model through intents

Standard language modeling approaches would train our dialogue model only to imitate the messages from our dataset but not to outperform them. To go beyond imitation learning, we made the dialogue model controllable by generating messages conditioned on a plan specified by the strategic reasoning module

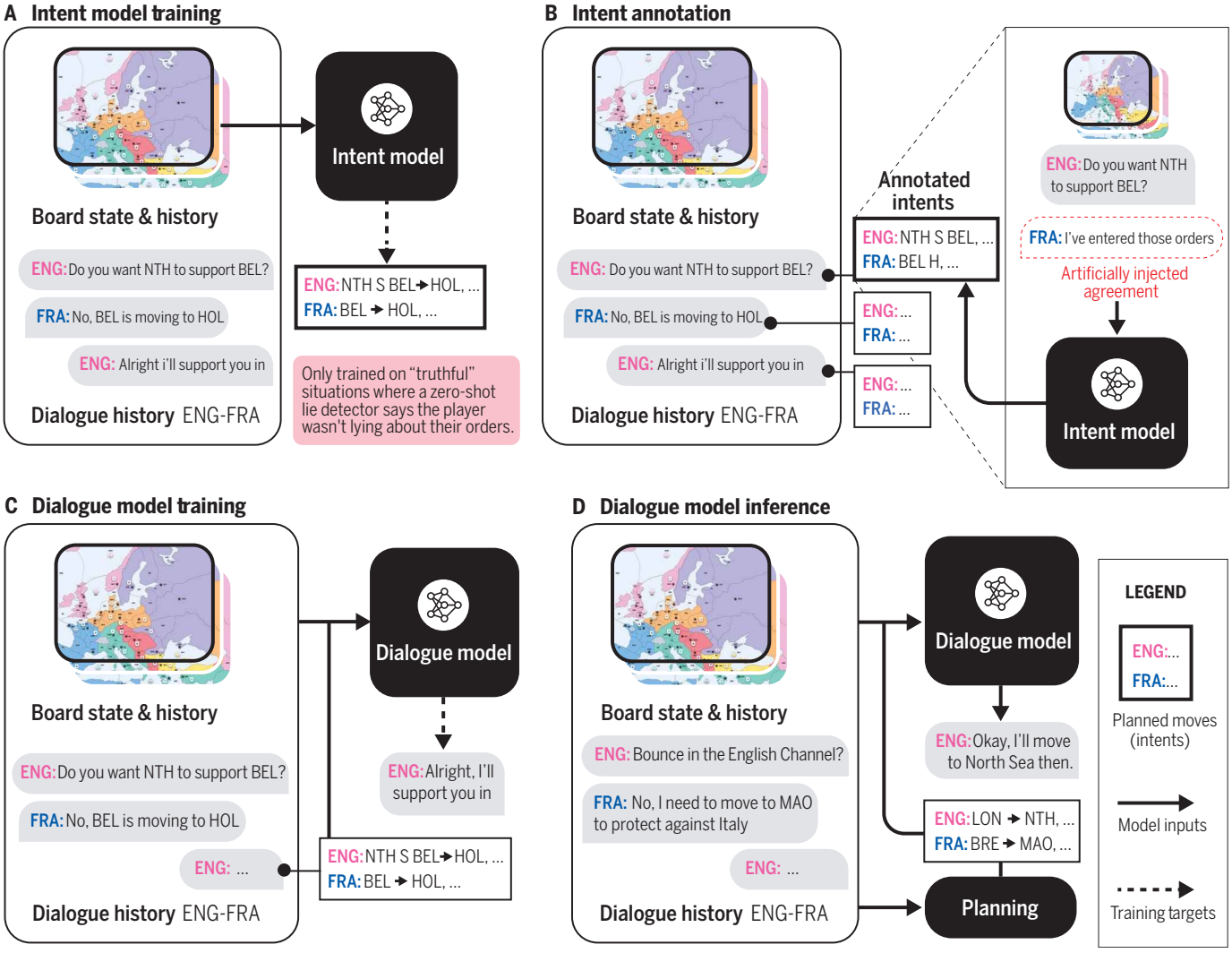


Fig. 2. Illustration of the training and inference process for intent-controlled dialogue. Actions are specified as strings of orders for units; for example, "NTH S BEL - HOL" means that North Sea will support Belgium to Holland. (**A**) An "intent model" was trained to predict actions for a pair of players on the basis of their dialogue. Training data was restricted to a subset in which dialogue is deemed "truthful" (SM, section D.2.3). (**B**) Each message in the dialogue training dataset was annotated with the output of the intent model on the dialogue up to that point, with an agreement message injected at the end. (**C**) The dialogue model was trained to predict each dataset message given the annotated intent for the target message. (**D**) During play, intents were supplied by the planning module instead.

(intents), resulting in higher-quality messages. More specifically, a message is defined to have intent $\mathbf{z}$ if $\mathbf{z}$ is the most likely set of actions that the sender and recipient will take—for both the current turn and several future turns—if no further dialogue occurs after the message is received. To establish this control, we developed techniques to automatically annotate every message in the training set with a set of actions corresponding to the message content. During training, the dialogue model learned the distribution $p_\theta\left[\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \mathbf{z}^{(i)}\right]$, where $\mathbf{z}^{(i)}$ represents the intent for datapoint $[\mathbf{x}^{(i)}, \mathbf{y}^{(i)}]$; as a result, at inference, time $\mathbf{z}$ provides a point of control over generation (*25*). We later describe the training and inference process, which is also illustrated in the pipeline in Fig. 2. The effect of the intents on the generated dialogue is demonstrated in Fig. 3; conditioning on different planned actions results in different messages.

We considered other notions of intent during development, such as controlling messages to focus on specific subsets of actions, third-party actions, or to have a particular tone. Richer intents are harder to annotate on human messages, are harder to select with the planning module, and create greater risk of taking the language model out of distribution.

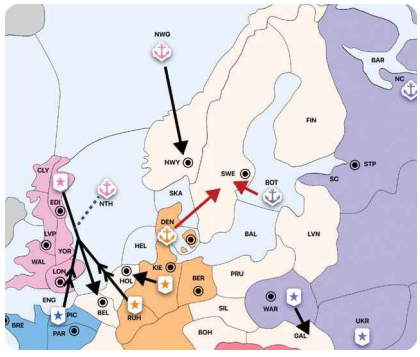
Annotating training messages with intents

When annotating messages in the training data with corresponding intents, our goal was for the proposed actions $\mathbf{z}^{(i)}$ to closely reflect the content of a message $\mathbf{y}^{(i)}$ so that at training time, the model learned to exploit the information in $\mathbf{z}^{(i)}$.

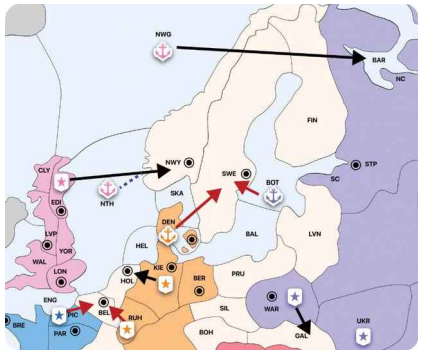
Naïvely, we could have used the actual actions played by the sender and recipient at the end of each turn in the span of the intent. However, these actions may not reflect the content of a message if (i) a message is not honest or (ii) subsequent messages change the sender's plans. To resolve (i), we predicted the most likely action according to a dialogue-conditional action prediction model trained on a "truthful" subset of the dataset, in which we predicted that a player's dialogue was not deceptive to others (SM, section D.2). This is showcased in Fig. 2A; we refer to this model as the intent model. To resolve (ii), we restricted the dialogue history that this intent model saw up to the message in question, which signaled to the model to predict actions as though the dialogue had ended at that point in time. We

Depicted intents:

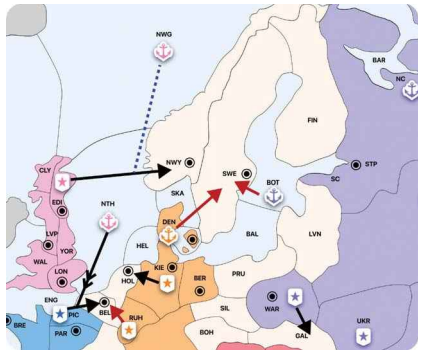
England convoys an army to Belgium with the support of France and Germany while taking Norway in a manner friendly to Russia.



England convoys an army to Norway and also aggressively moves to the Barents Sea against Russia.



England convoys an army to Norway and supports France to gain Belgium, preventing Germany from taking Belgium.



Example generations from the dialogue model with these actions as the conditioned intent:

ENG→ FRA Mind supporting Edi - Bel?

ENG→ GER Do you wanna support my convoy to Bel? With Italy going aggressive France will fall quickly and we can make gains off of both Russia and France.

ENG→ RUS How are you thinking Germany is gonna open? I may have a shot at Belgium, but I'd need your help into Den next year.

ENG→ FRA What's the plan for Bel?

ENG→ GER I will be convoying my army to Norway and moving to Barrents Sea.

ENG→ RUS Hey, I moved to Edi to keep my options open. Do you have a problem with me putting an army in Nwy?

ENG→ FRA Let me know if you want me to support you to Bel, otherwise I'll probably poke Hol.

ENG→ GER Looks like you'll get three builds unless France bounces you! Are you gonna bounce Russia or not?

ENG→ RUS Hey, I moved to Edi to keep my options open. Do you have a problem with me putting an army in Nwy?

Fig. 3. The effect of intents on Cicero's dialogue. Pictured are three different possible intents in the same game situation. In each case, we show a message generated by Cicero (England; pink) to France (blue), Germany (orange) and Russia (purple) conditioned on these intents. Each intent leads to quite different messages, which are consistent with the intended actions.

additionally added messages to the dialogue history that suggested a conclusive agreement between the two parties (Fig. 2B). As a result, we obtained a high degree of correspondence between the action annotated as the intent of a message and the content, achieving a score of 97% on a small test set designed to measure this correspondence (compared with 77% for a simpler baseline) (table S2). Then, the dialogue model could be trained in the manner described in the above section Imitation dialogue model and in Fig. 2C (SM, section D.2).

Selecting intents during play

During play, Cicero used the strategic reasoning module to select intent actions for the current turn (Fig. 2D), whereas intent actions for future turns were generated by means of a human-imitation model.

Agent intent action for current turn

Cicero conditioned its dialogue on the action that it intends to play for the current turn. This choice maximizes Cicero's honesty and its ability to coordinate but risks leaking information that the recipient could use to exploit it (for example, telling them which of their territories Cicero plans to attack) and sometimes led to out-of-distribution intents when the intended action was hostile, because in adversarial situations, humans may rarely communicate their intent honestly. We describe approaches for mitigating these risks in the section Message filtering.

	DIALOGUE QUALITY RATINGS (%)			
	Consistent with state	Consistent with plan	High quality	Perplexity
Language model	61.90	76.19	20.64	8.02
+ game state grounding	84.13	83.33	29.37	7.94
+ intent grounding (CICERO)	87.30	92.86	37.30	7.70

Fig. 4. Controllable dialogue modeling results. We report dialogue quality ratings and perplexity on the validation set for the Cicero dialogue model and compare them with a baseline without intent grounding and a baseline without either intent or game-state grounding ("Language model"). Dialogue quality ratings were calculated according to expert annotation of generated messages in 126 situations; we report the percent of messages (before filtering) labeled as consistent with the game state, as consistent with the plan for the next actions, and as particularly high quality. Lower perplexity corresponds to more probability mass on the ground-truth human messages.

Recipient intent action for current turn

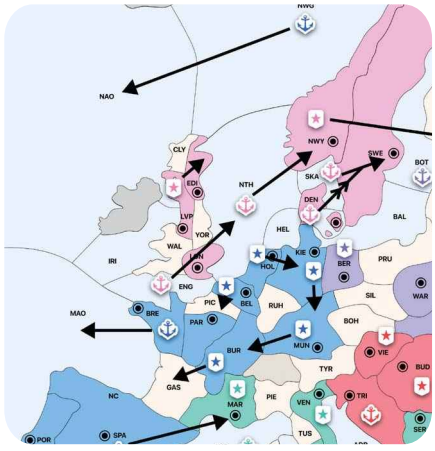
Cicero considered the subset of recipient actions with high likelihood under its beliefs about their policy. High likelihood requires that either an action is deemed beneficial for the recipient and/or that they are believed to be likely to play it given the dialogue. Among this restricted set, Cicero selected the recipient action with the highest expected value for itself (SM, section D.2.4).

Dialogue modeling results

We compared the performance of our dialogue model with a baseline without intent grounding and one without intent or game-state grounding (a "language model"). We report

England agrees:

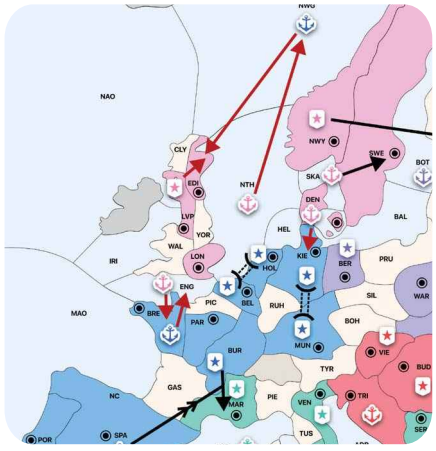
ENG → FRA Yes! I will move out of ENG if you head back to NAO.



Cicero predicts England will retreat from ENG to NTH 85% of the time, backs off its own fleet to NAO as agreed, and begins to move armies away from the coast.

England is hostile:

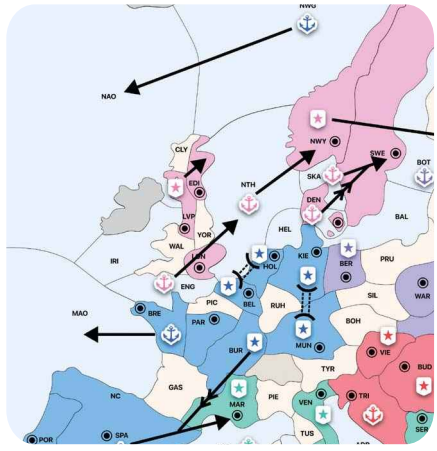
ENG → FRA You've been fighting me all game. Sorry, I can't trust that you won't stab me.



Cicero does not back off its fleet but rather attacks EDI with it, and leaves its armies at the coast to defend against an attack from England, predicting that England will attack about 90% of the time.

England tries to take advantage of Cicero:

ENG → FRA Yes! I'll leave ENG if you move KIE -> MUN and HOL -> BEL.



Strategic planning rejects the possibility of vacating KIE and HOL, because it would make Cicero too vulnerable. Cicero backs off its fleet to NAO but keeps armies at the coast to defend.

Fig. 5. The effect of dialogue on Cicero's strategic planning and intents. Cicero (France; blue) and England (pink) are entangled in a fight, but it would be beneficial for both players if they could disengage. Cicero has just messaged England "Do you want to call this fight off? I can let you focus on Russia and I can focus on Italy." Pictured are three ways that England might reply and how Cicero adapts to each. (**Left** and **middle**) Because Cicero's planning anchors around a dialogue-conditional policy model, its predictions for other players and accordingly its own plans are flexible and responsive to negotiation with other players. (**Right**) Yet Cicero also avoids blindly trusting what other players propose by rejecting plans that have low predicted value and run counter to its own interests.

both perplexity on the validation set and dialogue quality rating scores, which were calculated on the basis of expert annotation of messages generated in 126 *Diplomacy* game situations. Experts were asked to label whether a message was (i) consistent with the game state, (ii) consistent with the agent's plan, and (iii) notably high quality, compared with that of an average human. Results are shown in Fig. 4, and more details regarding this evaluation are provided in the SM, section D.2.3. Our model outperformed the baselines on all metrics. The improvement in validation perplexity demonstrated that the model can use additional grounding information to better predict human messages. Expert annotations showed that the grounding information provided by the intents and game state led to higher-quality messages that were highly consistent with the agent's intended action.

Strategic reasoning

To generate the intents for dialogue and to choose the final actions to play each turn, Cicero ran a strategic reasoning module that predicts other players' policies (a probability distribution over actions) for the current turn according to the state of the board and the shared dialogue and then chose a policy for itself for the current turn that responded optimally to the other players' predicted policies.

Doing this with human players requires predicting how humans will play. A popular approach in cooperative games is to model the other players' policies through supervised learning on human data, which is commonly referred to as behavioral cloning (BC). However, pure BC is brittle, especially because a supervised model may learn spurious correlations between dialogue and actions (fig. S6).

To address this problem, Cicero used variants of piKL (*26*) to model the policies of players. piKL is an iterative algorithm that predicts policies by assuming each player i seeks to both maximize the expected value of their policy π_i and minimize the Kullback-Leibler (KL) divergence between π_i and the BC policy, which we call the anchor policy τ_i. An anchor strength parameter $\lambda \in [0, \infty)$ trades off between these competing objectives.

piKL: KL-regularized planning

piKL is an iterative algorithm that predicts player policies. A complete description of the algorithm can be found in the SM, section E.1. piKL treats each turn in *Diplomacy* as its own subgame in which each player i simultaneously chooses an action a_i that results in joint action $a = (a_1, ..., a_n)$, and then each player i receives a reward $u_i(a)$ determined by a value function u_i. We discuss the training of this value function later below.

piKL assumes player i seeks a policy π_i that maximizes the modified utility function

$$U_i(\pi_i, \pi_{-i}) = u_i(\pi_i, \pi_{-i}) - \lambda D_{KL}(\pi_i \| \tau_i) \quad (1)$$

where π_{-i} represents the policies of all players other than i, and $u_i(\pi_i, \pi_{-i})$ is the expected value of π_i given that other players play π_{-i}. Specifically, let $Q_i^{t-1}(a_i) = u_i\left(a_i, \pi_{-i}^{t-i}\right)$ and let

$$\pi_i^{\Delta t}(a_i) \propto \tau_i(a_i) \exp\left[\frac{Q_i^{t-1}(a_i)}{\lambda}\right] \quad (2)$$

On each iteration t, piKL updates its prediction of the players' joint policies to be

$$\pi^t = \left(\frac{t-1}{t}\right)\pi^{t-1} + \left(\frac{1}{t}\right)\pi^{\Delta t} \quad (3)$$

piKL provably converges to an equilibrium in the modified utility space (*26*). When the anchor strength λ is set to a large value, piKL predicts that player i's policy will be close to the anchor policy τ_i. When λ is small, piKL predicts that player i's policy will have high expected value and may deviate substantially from τ_i.

A generalization of piKL referred to as Distributional Lambda piKL (DiL-piKL) replaces the single λ parameter in piKL with a probability distribution over λ values (SM, section E.1.3). On each iteration, each player samples a λ value from their distribution. In practice, we found this led to better performance (*17*).

Dialogue-conditional planning

Because dialogue influences the BC policy (the anchor policy τ_i), piKL provides a mechanism for dialogue to influence policy predictions. Different possible messages between Cicero and another player may produce different anchor policies (Fig. 5), which ultimately gives different final predictions about what that player will do.

Other players may of course be deceptive about their plans. Cicero does not explicitly predict whether a message is deceptive or not but rather relies on piKL to directly predict the policies of other players on the basis of both the BC policy (which conditions on the message) and on whether deviating from the BC policy would benefit that player.

Because dialogue in *Diplomacy* occurs privately between pairs of players, Cicero must reason about what information players have access to when making predictions. For example, if Cicero is coordinating an attack with an ally against an adversary, Cicero's prediction of the adversary's policy must account for the adversary not being aware of the intended coordination. Cicero accomplished this by predicting by means of pairwise piKL what every other player's policy will be.

Specifically, during strategic planning, for each player j, Cicero computed an anchor policy for both itself and player j on the basis of their shared conversation, the board state, and the recent action history. Cicero then ran DiL-piKL for the two players to predict player j's policy. On each iteration, Cicero assumed that the remaining five players would play according to a policy computed by means of RL, conditional on the policies of Cicero and player j. This process gave an independent prediction of each player's policy.

Next, Cicero accounted for the players' policies not being independent owing to their ability to correlate their actions through private dialogue that Cicero did not observe. Cicero accomplished this by constructing an approximate joint policy for all other players through self-normalized importance sampling: We sampled $N = 1000$ joint actions a from the independent piKL policies of the other players and reweighted them by the likelihood ratio of a under the correlated and independent RL policies, respectively.

Last, Cicero chose the action a_i that best responds to the predicted joint policy π_{-i} of the other players, while still being as consistent as possible with its dialogue. Specifically, Cicero chose the action $\text{argmax}_{a_i} u_i(a_i, \pi_{-i}) + \lambda \log \tau_i(a_i)$, where u_i is the RL value function, $\tau_i(a_i)$ is the probability of the action under the dialogue-conditional imitation policy, and $\lambda = 3 \times 10^{-3}$. Cicero used a smaller λ for regularizing its best response than for its computation of other players' policies; thus, the dialogue more strongly informed Cicero's expectations of how other players would coordinate while still allowing Cicero more leeway to deviate when the action that it predicted humans would most likely choose in its situation was suboptimal.

Self-play RL for improved value estimation

Applying piKL requires a state value function. Self-play provides an avenue for training such a value function but risks becoming incompatible with human play (*16*, *17*). To address this, we used piKL during self-play to keep the policies human-compatible.

One challenge in doing self-play in *Diplomacy* is that players may adapt their actions substantially on the basis of dialogue with other players, including coordinating joint actions. Explicitly simulating conversations would be extremely expensive in RL. However, a key insight is that a joint, shared BC policy trained on the joint action distribution of the human data already implicitly captures the effects of dialogue on the action distribution of human players by modeling that action distribution directly.

We therefore developed Correlated and Shared (CoShar) piKL, which allowed for regularization toward a joint, correlated anchor policy τ shared by all players rather than toward per-player policies. In this way, we relied on the joint anchor policy to capture the correlation between all players' policies. Specifically, CoShar piKL differs from standard piKL in that the probability of joint action $a = (a_1, ..., a_n)$ in policy $\pi^{\Delta t}$ becomes

$$\pi^{\Delta t}(a) \propto \tau(a) \exp\left[\sum_{i \le n} \frac{Q_i^{t-1}(a_i)}{\lambda}\right] \quad (4)$$

We found that CoShar piKL retained much of the correlation present in the joint anchor policy τ while also modeling strong human play better than imitation alone.

Our resulting self-play algorithm operated similarly to AlphaZero (*27*) and ReBeL (*28*) by applying planning "in the loop" as the improvement operator for RL. In our case, planning was by use of an approximated version of CoShar piKL. We generated self-play trajectories in which on each turn we computed the CoShar piKL policy using a learned state-value model. We regressed the joint policy model toward that policy and regressed the value model toward the expected values of all players under that policy. We then sampled a joint action from that policy to generate the next state in the trajectory. The anchor policy was fixed throughout training to anchor the RL near human play (SM, section E.4).

Message filtering

Prior work has shown that neural language models suffer from contradictions and inconsistency as well as a tendency to "hallucinate," or generate factually incorrect information (*29*). In the complex domain of *Diplomacy*, dialogue models exhibit both these problems and other more subtle mistakes, such as deviations from the intents used to control the message or blunders in the strategic content of the message. We approached this problem by filtering generated messages using a series of classifiers and checks to detect common issues. We outline

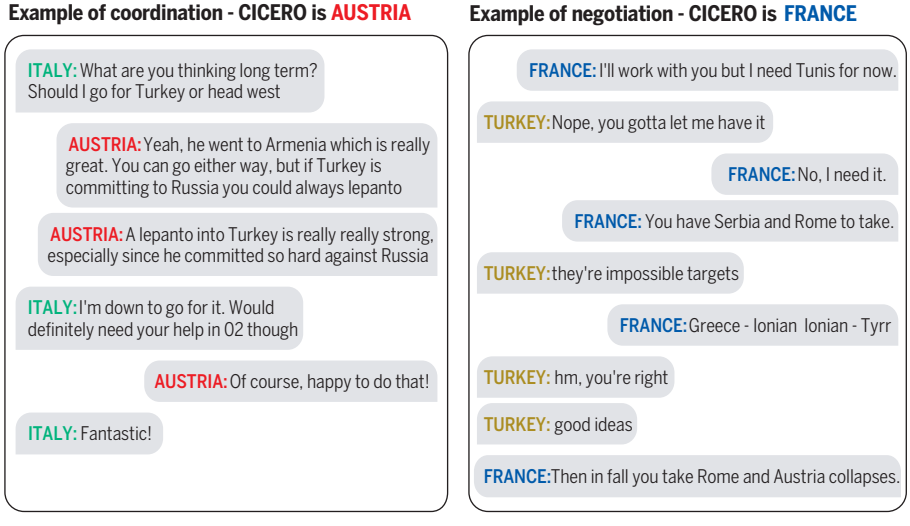


Fig. 6. Successful dialogue examples. Examples of Cicero (**left**) coordinating and (**right**) negotiating with authors of this paper in test games.

several of these filters here, with additional details in the SM, section D.3.

Discriminating between human text and counterfactuals

Much work has used adversarial or counterfactual examples to improve the robustness of natural language systems (*30*, *31*). Following this approach, we generated many kinds of counterfactual messages that contained mistakes that language models are prone to, including heuristically corrupted text as well as model-generated negatives. We trained a suite of 16 classifiers to discriminate between the ground-truth human message and different kinds of counterfactual messages (sometimes varying the random seed or context information available) and used these classifiers in an ensemble to filter messages. This approach risked overly filtering complex messages that contain precise plans and accepting bland messages, such as "ok," which are unlikely to contain mistakes. However, we found that carefully designing our ensemble allowed us to filter most nonsensical messages with minimal impact on message complexity: On a small evaluation set with 362 expert-annotated examples, we found that we could detect 83% of nonsense messages, without substantial impact to message diversity as measured by the proxy of message length and the number of references to *Diplomacy*-specific entities (SM, section D.3.1).

Intent correspondence

As noted previously, controlling dialogue generation through intents has the twofold benefit of improving the strategic value of a message and reducing discussion of impossible moves or other hallucinations. However, this control is imperfect, and the dialogue model may generate messages that contradict the intents it conditions on. To address this, we filtered messages that would reduce the likelihood of the actions in the intent. Evaluating this method on a small test set of 1013 expert-annotated messages, we achieved a recall of 65%, filtering 24% of all messages (SM, section D.3.2).

Value-based filtering

Conditioning on intents can lead to "information leakage," in which the agent reveals compromising information about its plan to an adversary (section Selecting intents during play). To mitigate this, we developed a method to score potential messages by their estimated value impact. We computed the piKL policies for all agents after each candidate message and filtered those that led to a lower expected value (EV) for Cicero playing its intended action. Expert evaluation on a set of 127 dialogue scenarios demonstrated that accepted messages were preferred over filtered messages 62% of the time ($P < 0.05$) (SM, section D.3.3).

Other filters

We additionally deployed other filters—for example, to detect toxic language (SM, section D.3.4)—and heuristics to curb bad behaviors, including repetition and off-topic messages (SM, section D.3.5).

Cicero in anonymous human play

Cicero participated anonymously in 40 games of *Diplomacy* in a "blitz" league on webDiplomacy.net from 19 August to 13 October 2022. This league played with 5-min negotiation turns; these time controls allowed games to be completed within 2 hours. Cicero ranked in the top 10% of participants who played more than one game and second out of 19 participants in the league that played five or more games. Across all 40 games, Cicero's mean score was 25.8%, which was more than double the average score of 12.4% of its 82 opponents. As part of the league, Cicero participated in an eight-game tournament that involved 21 participants, six of whom played at least five games. Participants could play a maximum of six games, with their rank determined by the average of their best three games. Cicero placed first in this tournament.

During games, players were not able to see the usernames of other players. Although webDiplomacy notifies users that the website has participated in AI research and that certain game modes allow users to play with AI agents, we evaluated Cicero in games with humans in which the participants were not explicitly informed that they were playing with an AI agent for that particular game. Cicero's participation as an AI was revealed to all players at the conclusion of the research (SM, section A.4).

Discussion

Cicero successfully combined strategic reasoning and dialogue to cooperate and negotiate with humans on a complex task, achieving strong human-level performance in the game of *Diplomacy*. Furthermore, Cicero passed as a human player for 40 games of *Diplomacy* with 82 distinct players, and no in-game messages indicated that players believed that they were playing with an AI agent. One player mentioned in post-game chat a suspicion that one of Cicero's accounts might be a bot, but this did not lead to Cicero being detected as an AI agent by other players in the league.

Two examples of coordination and negotiation are shown in Fig. 6. In the coordination example, we observed Cicero building an alliance through discussion of a longer-term strategy. In the negotiation example, Cicero successfully changed the other player's mind by proposing mutually beneficial moves. Despite dishonesty being commonplace in *Diplomacy*, we were able to achieve human-level performance by controlling the agent's dialogue through the strategic reasoning module to be largely honest and helpful to its speaking partners.

Although Cicero is shown to be effective at cooperating with humans, it occasionally sent messages that contained grounding errors, contradicted its plans, or were otherwise strategically subpar. Although we reduced errors with a suite of filters, *Diplomacy* poses an interesting benchmark for studying this problem. We suspect that these mistakes did not raise further suspicions that Cicero was an AI agent because of the time pressure imposed by the game, as well as because humans occasionally make similar mistakes. As such, formats of *Diplomacy* with longer negotiation periods could provide an even further challenge for future work because players typically engage in more detailed and complex negotiation in these formats.

From a strategic perspective, Cicero reasoned about dialogue purely in terms of players' actions for the current turn. It did not model how its dialogue might affect the relationship with other players over the long-term course of a game. Considering this might allow it to deploy dialogue more strategically. Furthermore, the expressive power of our intent representation limited Cicero's ability to control richer affordances of dialogue such as strategically revealing information, asking questions, or providing explanations for its actions. There remain many open problems for intentional use of dialogue, and *Diplomacy* provides a rich testbed to explore these connections between strategy and communication, with the goal of improving coordination between humans and agents.

Ethical considerations

We discuss ethical considerations for this research further in the SM, including privacy considerations for data usage (SM, section A.1), potential harms resulting from toxic or biased language generation (SM, section A.2), avenues for misuse of goal-oriented dialogue technology (SM, section A.3), and AI agent disclosure to human players (SM, section A.4).

REFERENCES AND NOTES

1. T. Brown *et al*., *Adv. Neural Inf. Process. Syst.* **33**, 1877 (2020).
2. M. Campbell, A. J. Hoane Jr., F. Hsu, *Artif. Intell.* **134**, 57–83 (2002).
3. D. Silver *et al*., *Nature* **529**, 484–489 (2016).
4. N. Brown, T. Sandholm, *Science* **365**, 885–890 (2019).
5. S. Kraus, D. Lehmann, Diplomat, an agent in a multi agent environment: An overview, in *IEEE International Performance Computing and Communications Conference* (IEEE Computer Society, 1988), pp. 434–435.
6. D. d. Jonge *et al*., The challenge of negotiation in the game of *Diplomacy*, in *International Conference on Agreement Technologies* (Springer, 2018), pp. 100–114.
7. P. Paquette *et al*., *Adv. Neural Inf. Process. Syst.* **32**, 4474–4485 (2019).
8. A. Dafoe *et al*., *Nature* **593**, 33–36 (2021).
9. M. Moravčík *et al*., *Science* **356**, 508–513 (2017).
10. N. Brown, T. Sandholm, *Science* **359**, 418–424 (2018).
11. O. Vinyals *et al*., *Nature* **575**, 350–354 (2019).

12. Dota 2 (*32*) is two-team zero-sum but with unlimited information sharing between teammates, which makes the game equivalent to 2p0s. Prior work found that self-play from scratch was sufficient for achieving superhuman performance in multiplayer poker (*4*), but this may be due to poker offering few opportunities for players to cooperate.
13. J. v. Neumann, *Math. Ann.* **100**, 295 (1928).
14. M. Lewis, D. Yarats, Y. Dauphin, D. Parikh, D. Batra, Deal or no deal? End-to-end learning of negotiation dialogues, in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (Association for Computational Linguistics, Copenhagen, Denmark, 2017), pp. 2443–2453.
15. A. P. Jacob, M. Lewis, J. Andreas, Multitasking inhibits semantic drift, in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2021), pp. 5351–5366.
16. A. Bakhtin, D. Wu, A. Lerer, N. Brown, *Adv. Neural Inf. Process. Syst.* **32**, 34 (2021).
17. A. Bakhtin *et al*., Mastering the game of no-press *Diplomacy* via human-regularized reinforcement learning and planning. arXiv:2210.05492 [cs.GT] (2022).
18. A. Holtzman, J. Buys, L. Du, M. Forbes, Y. Choi, The curious case of neural text degeneration, *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* (OpenReview.net, 2020).
19. S. Keizer *et al*., Evaluating persuasion strategies and deep reinforcement learning methods for negotiation dialogue agents, in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers* (Association for Computational Linguistics, 2017), pp. 480–484.
20. T. Hiraoka, G. Neubig, S. Sakti, T. Toda, S. Nakamura, Construction and analysis of a persuasive dialogue corpus, in *Situated Dialog in Speech-Based Human-Computer Interaction* (Springer, 2016), pp. 125–138.
21. X. Wang *et al*., Persuasion for good: Towards a personalized persuasive dialogue system for social good, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Association for Computational Linguistics, Florence, Italy, 2019), pp. 5635–5649.
22. K. Shuster *et al*., Language models that seek for knowledge: Modular search and generation for dialogue and prompt completionarXiv:2203.13224 (2022).
23. A. Vaswani *et al*., Attention is all you need, in *Advances in Neural Information Processing Systems*, I. Guyon, *et al*., eds. (Curran Associates, Inc., 2017), vol. 30.
24. M. Lewis *et al*., BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10,2020*, D. Jurafsky, J. Chai, N. Schluter, J. R. Tetreault, Eds. (Association for Computational Linguistics, 2020), pp. 7871–7880.
25. N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, R. Socher, CTRL: A conditional transformer language model for controllable generation. arXiv:1909.05858 [cs.CL] (2019).
26. A. P. Jacob *et al*., Modeling strong and human-like gameplay with KL-regularized search, in *International Conference on Machine Learning* (PMLR, 2022), pp. 9695–9728.
27. D. Silver *et al*., *Science* **362**, 1140–1144 (2018).
28. N. Brown, A. Bakhtin, A. Lerer, Q. Gong, *Adv. Neural Inf. Process. Syst.* **33**, 17057 (2020).
29. Z. Ji *et al*., Survey of hallucination in natural language generation. arXiv:2202.03629 [cs.CL] (2022).
30. P. Gupta, Y. Tsvetkov, J. P. Bigham, Synthesizing adversarial negative responses for robust response ranking and evaluation, in *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6,2021*, C. Zong, F. Xia, W. Li, R. Navigli, Eds. (Association for Computational Linguistics, 2021), vol. ACL/IJCNLP 2021 of *Findings of ACL*, pp. 3867–3883.
31. M. Alzantot *et al*., Generating natural language adversarial examples, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31–November 4,2018*, E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii, eds. (Association for Computational Linguistics, 2018), pp. 2890–2896.
32. C. Berner *et al*., Dota 2 with large scale deep reinforcement learning. arXiv:1912.06680 (2019).
33. FAIR *et al*., Supplementary data for "Human-level play in the game of *Diplomacy* by combining language models with strategic reasoning". *Zenodo* (2022).
34. FAIR *et al*., Code for "Human-level play in the game of *Diplomacy* by combining language models with strategic reasoning". *GitHub* (2022); https://github.com/facebookresearch/diplomacy_cicero.

ACKNOWLEDGMENTS

We thank K. Kuliukas for providing access to the WebDiplomacy data and for supporting this research; J. Andreas, N. Goyal, P. Paquette, K. Shuster, and S. Zhang for helpful support and discussions; and J. Weston for feedback on early drafts of this paper. **Funding:** All funding was provided by Meta. **Author contributions:** Authors are listed alphabetically in the byline. A.B., N.B., E.D., G.F., C.F., D.F., J.G., H.H., A.P.J., M.Ko., M.Kw., A.L., M.L., A.R., S.R., W.S., A.W., D.W., and H.Z. contributed to the development of Cicero algorithms, code, and experiments. A.G., K.K., and M.Z. provided *Diplomacy* expertise and data annotation. S.M. and D.R. contributed to data collection. A.H.M. and J.S. managed the research team. A.B., N.B., E.D., G.F., C.F., D.F., J.G., A.P.J., A.L., M.L., A.H.M., A.R., W.S., A.W., and D.W. wrote the paper. **Competing interests:** None declared. **Data and materials availability:** The figure and table data are deposited in Zenodo (*33*). The codebase is available at (*34*). Training data was licensed from WebDiplomacy.net by Meta AI.

SUPPLEMENTARY MATERIALS

science.org/doi/10.1126/science.ade9097
Materials and Methods
Figs. S1 to S10
Tables S1 to S15
References (*35–90*)

Submitted 15 September 2022; accepted 9 November 2022
10.1126/science.ade9097