

Forecasting AI Development with Arms Race Models

Andy Liu

Project advisor: Christina Edholm

Abstract. Forecasting the likelihood that Transformative Artificial Intelligence (TAI) will be developed has become increasingly pertinent for AI scholars. However, such models have attracted comparatively less attention from the applied mathematics community, leaving space for further modeling contributions. In this paper, we extend Cotra’s TAI forecast with Hamblin’s model of arms races, allowing us to analyze the impact of competitive dynamics between the United States and China on the likelihood that TAI is developed. We also implement an extension of Hamblin’s model to handle tripolar (three-way) arms races and similarly integrate this into our model. We find that bipolar arms race dynamics will accelerate short-term AI development and that tripolar arms race dynamics will accelerate long-term AI development relative to baseline models.

1. Introduction. Transformative Artificial Intelligence (TAI) is any artificial intelligence that precipitates a transition comparable to (or more significant than) the agricultural or industrial revolution [6]. Forecasting the development of TAI has gained significant attention in some circles, which seek to craft policy solutions to ensure that such technologies can be deployed safely and beneficially. One such model in this space was introduced by Cotra [1], who sought to model the growth of AI computing capabilities over time. They used biological anchors in human thought to consider how much computing power would be necessary to create human-level artificial intelligence. They then estimated the amount of spending it would take to gain this level of computing power, creating a model that predicted when sufficient resources would be spent on AI development to achieve this level.

However, when forecasting future AI spending, Cotra assumed a constant rate of growth in spending from “frontier” countries such as China and the United States. In reality, such

nations engage in competitive behavior over AI development [2], which could lead to “arms race“-like spending dynamics. In [3], Gray defines an “arms race“ as a scenario where rival states spend increasing amounts of resources on their militaries in order to defend against each other. While spending on TAI development is not necessarily military, the defense and economic implications of TAI, as well as the competitive atmosphere in TAI development, are sufficient to allow us to model it as such. However, while AI spending is currently dominated by the United States and China, continued AI development could attract further competition from other actors, such as the European Union. This motivates us to also consider multipolar arms race dynamics, and how they might affect Cotra’s Model (either similarly to, or differently from, a bipolar model).

We are specifically interested in answering the following questions:

1. How does modeling AI spending as an arms race change TAI forecasting models’ projections?
2. What are the differences in spending dynamics between bipolar and multipolar arms races?

While the classic Richardson Arms Race model introduced in [8] is the most famous example of an arms race model, we instead consider Hamblin’s asymmetric model [4]. In this paper, Hamblin discusses a process by which his model can be fit to existing data; by instead fitting it to existing data on AI spending between American and Chinese organizations, we build an “arms race” model of AI spending. We also extend this model to consider the dynamics of a tripolar arms race between the United States, China, and the European Union. Finally, we analyze the equilibrium behavior of our new models. We find that both arms race models predict more rapid near-term growth in AI spending and that the United States will remain the world leader in AI development. However, while the bipolar model reaches an equilibrium state relatively quickly, the tripolar model predicts more sustained growth in AI spending, outpacing both the bipolar and Cotra’s original model over a 20-year period. This

is also verified by our equilibrium analysis, which suggests that the increased competition exhibited in a tripolar system leads to higher AI spending and increases the likelihood of TAI development over time.

2. Methods.

2.1. Assumptions. In order to incorporate an arms race element into Cotra’s TAI model forecast, certain assumptions are needed to guarantee the feasibility of this approach:

1. (Arms Race) First, we need to assume that arms race models will accurately capture the dynamics of future AI spending between countries. This at least somewhat accurately reflects reality; in [2], Ding notes that American AI strategy relies on “defending [its] lead” from nations such as China, while Chinese AI strategy is heavily responsive to American AI advancements.
2. (Bipolarity/Tripolarity) Next, we assume that AI development will remain dominated by a small number of national or supernational actors; namely the United States, China, and (in our tripolar model) the European Union. In their Harvard Business Review article, [7] note that the United States and China do indeed dominate global AI research, with Europe also heavily involved.
3. (Scaling) Finally, we assume the rate at which the actors included in our arms race model will privately spend on AI development will grow at the same rate as total compute spending. Because our model separates spending between the United States, China, and the European Union, we needed to use a different source of data than the Cotra model. We align these two datasets by “scaling up” our initial total spending to be equal to that of the Cotra model. This allows us to study the relative growth of AI spending within the two models by giving them the same starting point.

2.2. Cotra’s Model of TAI Development. We consider a model of Transformative AI Development introduced by [1]. This aims to predict the likelihood of it being plausible to train Transformative AI by a certain year based on underlying assumptions about the amount

of computing power required. Cotra models this with a spreadsheet¹ which is based off a hypothesis chart defining $P_{\text{TAI}}(C)$, how likely TAI is to be trained given a certain amount of computing power. This probability chart forms a logistic curve.

The model is centered around two variables. The first is $F(t)$, which represents how efficiently nations can convert AI spending into computing power (as measured in FLOPs, or floating-point operations per second, per dollar of spending) in the year t . $F(t)$ is set to grow exponentially, doubling every 2.5 years according to Moore’s Law until reaching a maximum efficiency threshold. The second is $W(t)$, or the extent to which researchers are willing to spend on AI in the year t . This is also set to grow exponentially based on the growth of global GDP and increased AI spending willingness, although Cotra does not cap this parameter as she does $W(t)$. Specifically,

$$F(t) = \min(F_0 \cdot 2^{\frac{t}{d}}, F_{\max})$$

$$W(t) = W_0 \cdot 2^{\frac{t}{d}} \cdot (1 + g)^t$$

Where the initial spending efficiency F_0 is $4 \cdot 10^{17}$, the maximum efficiency F_{\max} is 10^{26} , the doubling rate d is equal to 2.5, the initial willingness to spend W_0 is equal to 10^9 , and the GDP growth rate g is equal to 0.03. The probability of achieving TAI by a certain year t is then simply $P_{\text{TAI}}(F(t) \cdot W(t))$.

However, because Cotra’s model assumes that spending will grow linearly exponentially over time, it ignores potential “arms race” dynamics between different nations in spending. Because artificial intelligence has important national security and economic implications, it is plausible that such race dynamics could emerge between great powers such as the United States in China. Thus, in this paper, we will try to incorporate an arms race model into Cotra’s TAI model.

¹[Linked here.](#)

2.3. Hamblin’s Asymmetric Model of Arms Races. [4] presents a discrete-time framing of the interaction process between nations that they show better reflects empirical arms race data. The model assumes that there are two nations in competition, that they have full control of their armament capacities, and that other actors do not directly influence military development. However, in contrast to Richardson’s concurrent model, Hamblin assumes that there will always exist a “leader”, denoted X , who is trying to preserve their advantage, and a “follower”, denoted Y , who is trying to catch up to the leader, in an arms race. Hamblin also models this dynamic as sequential, by having the follower act first with the leader reacting to the follower’s armament development. This paper also subscribes to a curvilinear model, based on empirical psychology research, where the change in psychological sensation (here, the desire to “catch up”) is a constant proportion of change in the related stimulus (here, the existing disparity in armaments). This constant is fitted to the empirical data that is later used to show the model’s fidelity to real-world arms race dynamics.

We use Hamblin’s power law formulation for both the leader and follower. While Hamblin suggests that the follower’s greater desire to catch up may lead to it responding exponentially to the leader’s spending, we use this simpler approach as it yields more approachable results on longer timeframes. Hamblin notes that the leader will feel insecurity proportional to a lagging indicator of the follower’s spending, and they will spend proportionally to their insecurity. This leads to the leader’s spending being equal to some power of a lagging indicator of the follower, while the follower will similarly respond to some power of the leader’s AI spending. Here, we use the previous year’s spending as the lagging indicator, which yields the following:

$$\begin{aligned}x(t+1) &= sy(t)^n \\ y(t+1) &= bx(t+1)^m\end{aligned}$$

Here, m and n are constants that control how “insecure” each actor feels about the other actor’s spending. s and b are both scaling constants and “grievance parameters” that represent how strongly each actor will respond to its insecurity by further spending on AI. Higher values of s and b should theoretically lead to faster arms race takeoffs.

2.4. A Tripolar Extension of Hamblin’s Model. We also present an extension of Hamblin’s arm race model to the case where there are three agents competing with each other to develop TAI. We formulate this model as having two followers, which modulate their spending according to that of the leader, and one leader, which reacts to the sum of the two followers’ spending. More formally, given a leader agent x and follower agents y and z , our update rule at time step $t + 1$ is as follows:

$$\begin{aligned}x(t + 1) &= s(y(t)^n + z(t)^n) \\y(t + 1) &= b_1 x(t + 1)^{m_1} \\z(t + 1) &= b_2 x(t + 1)^{m_2}\end{aligned}$$

Here, the two followers are given separate sets of parameters to model differing grievance levels between nations, while the leader’s parameters remain the same as in Hamblin’s original formulation.

The usage of the sum of the other agents is motivated by similar extensions of other arms race models. For example, [5] presents a tripolar extension of Richardson’s Arms Race Model where each agent responds to the sum of the other two agents’ spending. Our model captures both this dynamic and the leader-follower setup of the original Hamblin model.

2.5. Connecting Hamblin’s and Cotra’s Models. Based off of our 2021 data, we use our Hamblin-type model to predict total AI spending from the United States, China, and the European Union from 2021 to 2040. Because Cotra uses the log of total AI spending in her

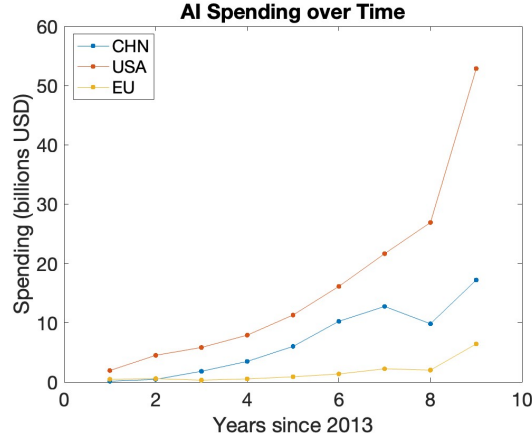


Figure 1. AI Spending over the years 2013-2021 from the US, China, and the EU, as measured by the Stanford AI Index. This data is used to fit our model parameters.

model, we compute the total predicted AI spending in our model over all actors. We then take the base-ten log of this time series and “scale up” all values in the predicted time series so that the initial spending levels align with that of the Cotra model.

We then exported this time series into the “willingness to spend” column in Cotra’s spreadsheet model. This represents the $W(t)$ time series formerly defined in subsection 2.2. Then, we analyzed the column containing the recalculated probability that the compute needed to train TAI is affordable by year based on the new spending dynamics, as well as Cotra’s original model. The computed projections for the **original model**, the **bipolar arms race model**, and the **tripolar arms race model** are linked here, with relevant columns bolded on the tab entitled “calcs sheet”.

2.6. Data Sourcing and Model Fitting. In order to estimate our model parameters for both the two- and three-nation arms races, we fit our models to data from the Stanford AI Index’s record of private AI spending [10]. This dataset has spending data from the United States, China, and European Union over the years 2013-2021, which we visualize in Figure 1. We can then use the model parameters that best fit national spending over this time period.

To fit our model parameters, we use the MATLAB function MultiStart [9]. This function

starts an instance of the `fmincon` local solver from a number of start points to find a global minimum of some objective function over a parameter space. We search over the range of parameters given in the Power Law section of Table 3 in [4], although large outliers are removed to improve the efficiency of the search. For the objective function, we use the candidate parameters to estimate 2014-2021 spending given 2013 spending and compute the sum of the Euclidean distances between predicted and actual spending over the period 2014-2021. This should theoretically give us the model parameters that best capture the behavior of each actor in this arms race.

3. Analysis and Results.

3.1. Model Fitting and Parameter Interpretation. Following the procedure outlined in subsection 2.6, we use MultiStart to fit parameters for bipolar and tripolar Hamblin models. We find that for the bipolar model, our model fitting procedure outputs the parameter values $s = 4.1273$, $n = 0.9332$, $b = 1.63$, and $m = 0.8633$. As shown in Figure 2, this produces reasonable approximations of both American and Chinese AI spending. However, our fitting process does have difficulty accounting for year-to-year noise such as the decrease in Chinese AI spending between 2019 and 2020. We note that s , the United States' grievance constant, is significantly higher than b , China's grievance constant. This suggests that the United States responds much more strongly to Chinese AI spending than China does to American AI spending.

We also follow the same steps to fit our tripolar model, which results in the parameters $s = 2.433$, $n = 1.0766$, $b_1 = 0.7824$, $m_1 = 0.7999$, $b_2 = 0.165$, $m_2 = 1.0137$. As shown in Figure 3, this produces reasonable approximations of American, Chinese, and European Union AI spending, although our models again have some difficulties fitting to volatile year-to-year changes in spending. Again, we can compare the grievance constants of all three actors. Doing so, we see that $s > b_1 > b_2$, suggesting that the United States still responds more strongly to Chinese and European Union AI spending than the other way around. Additionally, China

FORECASTING AI DEVELOPMENT WITH ARMS RACE MODELS

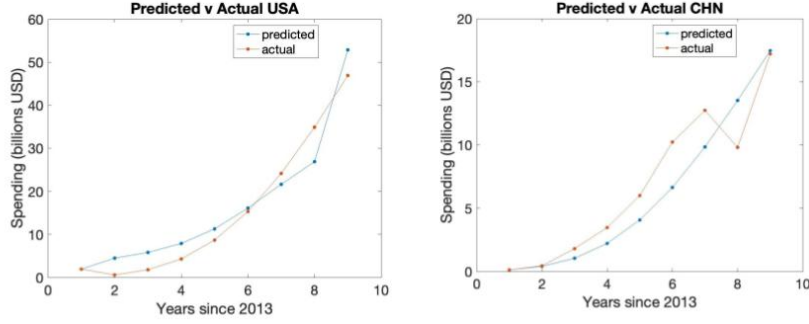


Figure 2. Our Hamblin model’s fit to American and Chinese spending on AI over 2013-2021, compared to the actual spending data from that time frame.

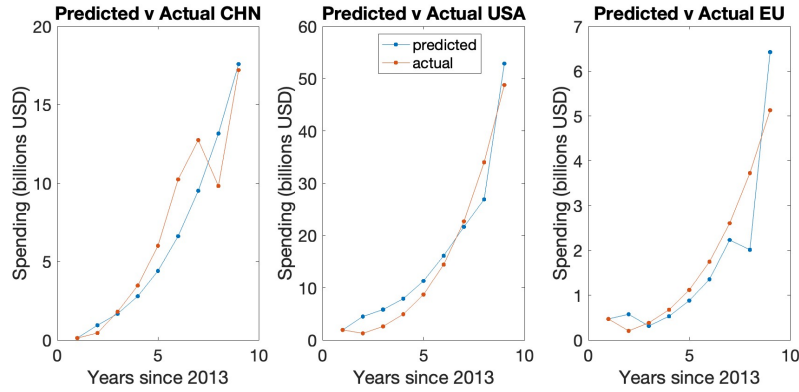


Figure 3. Our tripolar Hamblin model’s fit to American, Chinese, and EU spending on AI over 2013-2021, compared to the actual spending data from that time frame.

responds more strongly to European Union AI spending than the other way around.

3.2. Model Predictions. With our bipolar and tripolar models fitted, we can then use both models to predict future spending on AI over the time period 2021-2040. We start by projecting AI spending in our bipolar model, the results of which are shown in [Figure 4](#). We find that American spending is predicted to continue to outpace Chinese spending by a factor of ≈ 2.5 . Additionally, following our bipolar arms race model, we find that spending levels are predicted to initially increase rapidly, before starting to converge toward the end of the twenty-year timeframe.

We can also perform the same analysis on our tripolar model’s spending predictions for 2021-2040. Unlike the bipolar model, the tripolar model does not show any signs of conver-

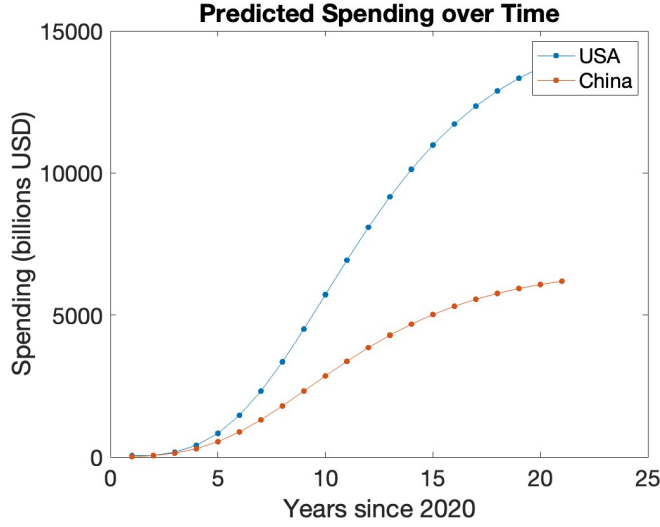


Figure 4. Our original Hamblin model’s predictions for American and Chinese spending on AI over the period 2021-2040.

gence in the given timeframe. Instead, American, Chinese, and European Union spending all appear to be growing exponentially over all twenty years (possibly because three-way arms races are inherently less stable than two-way arms races due to the increased number of actors). As a consequence of these accelerated arms race dynamics, we also find that net spending in 2040 is predicted to be significantly higher in the tripolar model than in the bipolar model. Finally, our tripolar model predicts that the European Union will actually overtake China in AI Spending in the year 2031. This is possibly due to the fact that while $b_1 > b_2$, $m_2 > m_1$, suggesting that the European Union’s spending is predicted to grow proportionally to a higher power of American AI spending than that of China.

Integration into Cotra’s Model. We now integrate our twenty-year predicted spending outcomes for both models into Cotra’s spreadsheet model, following the procedure outlined in subsection 2.5. In Figure 6, we compare our bipolar and tripolar arms race spending predictions to Cotra’s original model of spending, which assumes constant growth in countries’ willingness to spend on AI. For ease of visualization, we plot the base-ten logarithm of net spending in all three models. We find that the bipolar arms race model projects very high

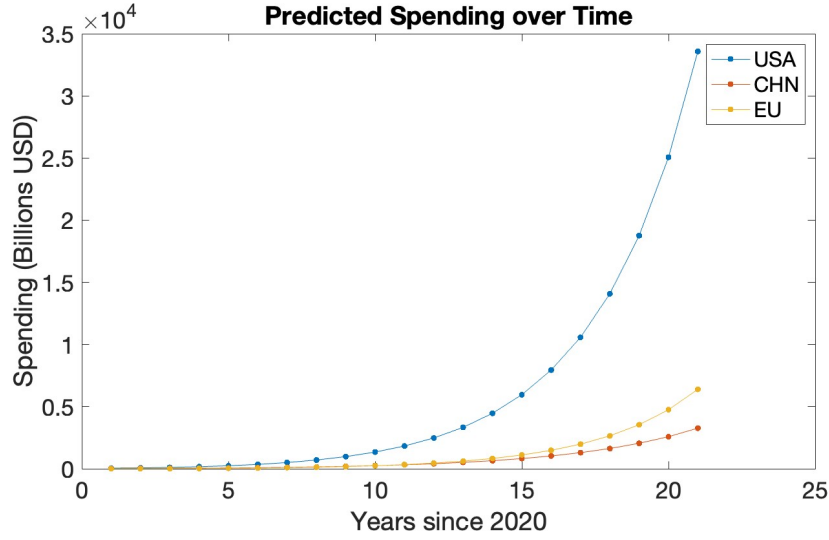


Figure 5. Our tripolar Hamblin model’s predictions for American, Chinese, and European Union spending on AI over the period 2021-2040.

initial growth in spending in the first decade of our predictions. However, this rapidly levels off as the bipolar model approaches an equilibrium state (to be discussed further in [subsection 3.3](#)). Meanwhile, the tripolar model also initially grows faster than Cotra’s exponential model, but unlike the bipolar model, it continues to grow exponentially over the entire timeframe. Around 2038, the tripolar model begins to predict the highest net spending of the three models, without any signs of imminent convergence.

These trends are also reflected when we compare the probabilities of being able to train TAI in the 2021-2040 timeframe that are predicted by each of the three models. As visualized in [Figure 7](#), the bipolar arms race’s early lead in spending means it predicts the highest probability of achieving TAI for every year before 2038. However, because its predicted spending levels begin to stabilize toward the end of this timeframe, it is again overtaken by the more dynamic tripolar model. Our tripolar model predicts a 42.36% probability of achieving the levels of computing power needed to train TAI by 2040, compared to 40.44% for the bipolar model and 38.39% for Cotra’s original model. These results somewhat validate Cotra’s base model, as all three models predict similar outcomes by the end of our twenty-year

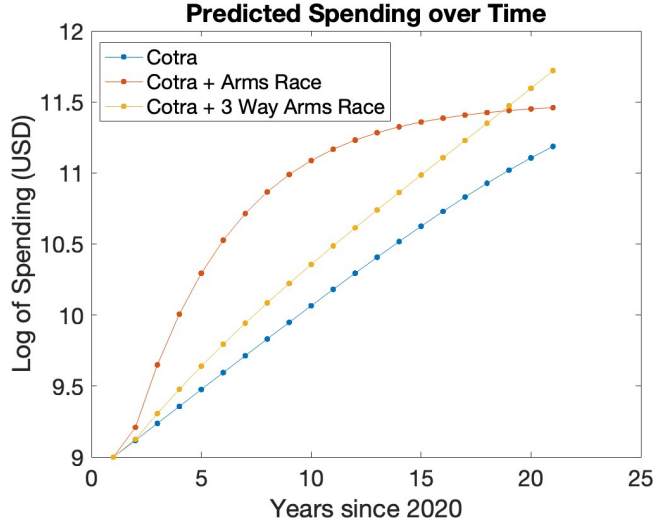


Figure 6. Our combined Hamblin+Cotra models' predicted total spending over the period 2021-2040, compared to the original Cotra model's prediction.

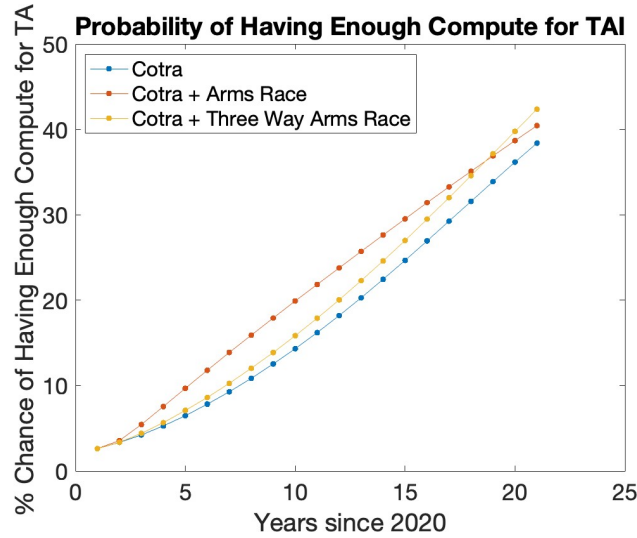


Figure 7. Our combined Hamblin+Cotra models' estimated probability of being able to train TAI over the period 2021-2040, compared to the original Cotra model's prediction.

period. However, the bipolar model displays different spending dynamics, predicting an initial surge in AI spending before the arms race stabilizes, while the tripolar model suggests that less stable arms race dynamics can further accelerate AI development.

3.3. Equilibrium Analysis.

Bipolar Model. We now seek to analyze equilibrium states for both our bipolar and tripolar combined models. Because $F(t)$ is capped at a constant value, it suffices to consider the steady-state of $W(t)$. To do so, we simply need to compute the fixed point of $x(t)$ in our Hamblin model, then use this to compute net spending at our model's equilibrium.

Note that at our fixed point, we must have $x(t+1) = x(t)$; we can then compute $x(t+1)$ in terms of $x(t)$ using the recurrence relation in [subsection 2.3](#). Doing so, we find that $x(t+1) = sy(t)^n = s(bx(t)^m)^n = sb^n x(t)^{mn}$. Thus, at our fixed point, we have $x(t) = sb^n x(t)^{mn} \Rightarrow x(t)(1 - sb^n x(t)^{mn-1}) = 0$. This means that either $x(t) = 0$ or $sb^n x(t)^{mn-1} = 1 \Rightarrow x(t) = (\frac{1}{sb^n})^{\frac{1}{mn-1}}$. Hamblin notes that the zero fixed point in this class of models is unstable in [\[4\]](#), so we consider only the nonzero fixed point. Thus, we have $x^*(t) = (\frac{1}{sb^n})^{\frac{1}{mn-1}}$ and (by direct substitution) $y^*(t) = (\frac{1}{s^m b})^{\frac{1}{mn-1}}$. Note that this implies that the leader will still retain the spending advantage at equilibrium if and only if $(\frac{1}{sb^n})^{\frac{1}{mn-1}} > (\frac{1}{s^m b})^{\frac{1}{mn-1}}$. This is true precisely when $mn > 1$ and $s^{m-1} > b^{n-1}$, or when $mn < 1$ and $b^{n-1} > s^{m-1}$.

Substituting our values of s , n , b , and m , computed in [subsection 3.1](#), we find that at equilibrium, we have $x^*(t) \approx 15360.48$ and $y^*(t) \approx 6703.64$. Note that these are approximately equal to the final spending levels forecasted by our model in [subsection 3.2](#), verifying our intuition that the model's predicted spending was very close to equilibrium by 2040. Scaling this to the Cotra model, we find that the steady-state value $W^*(t)$ is equal to $\frac{10^9}{x(0)+y(0)}(x^*(t) + y^*(t)) \approx 4.17 \cdot 10^{11}$. This means that the probability TAI will be achievable at the steady state of our arms race is $P_{\text{TAI}}((F_{\text{max}}(t) \cdot W^*(t))) \approx 0.61$. In practice, our model's approach to $W^*(t)$ will largely depend on the rate at which $F(t)$ increases, given the closeness of $W(20)$ to the equilibrium value $W^*(t)$ (which can be seen in [Figure 4](#), in which $x(20) \approx x^*(t)$).

Tripolar Model. To simplify our analysis, we assume that $m_1 \approx m_2$ and replace both with a single variable m . This is motivated by the belief that the two followers y_1 and y_2 should have relatively similar feelings of insecurity towards the leader's spending. However, we still assume that the two followers have different scaling/grievance constants to model differences

in their behavior. Further analysis should consider the impact of significantly different m_1 and m_2 values.

Again, we seek to first compute the fixed point value $x^*(t)$ by setting $x(t+1) = x(t)$ and using [subsection 2.4](#) to solve for $x(t)$. Doing so, we find that $x(t+1) = s(y(t)^n + z(t)^n) = s(b_1x(t+1)^m)^n + s(b_2x(t+1)^m)^n$. Thus, similarly to our analysis in [subsection 3.3](#), $x(t)(1 - (sb_1^n + sb_2^n)x(t)^{mn-1}) = 0$. We then consider only the nonzero fixed point, finding that $x^*(t) = (\frac{1}{sb_1^n + sb_2^n})^{\frac{1}{mn-1}}$. Substituting this value into our formulas for $y(t)$ and $z(t)$, we directly compute $y^*(t) = b_1(\frac{1}{sb_1^n + sb_2^n})^{\frac{m}{mn-1}}$ and $z^*(t) = b_2(\frac{1}{sb_1^n + sb_2^n})^{\frac{m}{mn-1}}$. This suggests that as long as both followers have the same m parameter, the relative spending of the two followers will be directly proportional to their grievance constants. Additionally, we find that (assuming without loss of generality that $b_1 > b_2$), $x^*(t)$ will be the largest of the three equilibrium spending values if and only if $(sb_1^n + sb_2^n)^{\frac{m-1}{mn-1}} > b_1^{\frac{m}{mn-1}}$. Note that the fitted parameters from [subsection 3.1](#) satisfy this condition, which is reasonable as our model predicts that the United States will maintain its spending advantage.

Substituting in the values $s = 2.433$, $n = 1.0766$, $b_1 = 0.7824$, $b_2 = 0.165$, $m = \frac{m_1+m_2}{2} = 0.9068$, we compute a fixed-point value of $x^*(t) = (\frac{1}{2.433 \cdot (0.7824^{1.0766} + 0.165^{1.0766})})^{\frac{1}{0.9068 \cdot 1.0766 - 1}} \approx 3.73 \cdot 10^{14}$. Following our approach in [subsection 3.3](#), we use this $x^*(t)$ value to compute a $x^*(t) + y^*(t) + z^*(t)$ value of $3.73 \cdot 10^{14} + y^*(t) + z^*(t) \approx 3.73 \cdot 10^{14} + 3.55 \cdot 10^{11} + 9.75 \cdot 10^{13}$. We then scale this to find that $W^*(t) = \frac{10^9}{x(0)+y(0)+z(0)}(x^*(t) + y^*(t) + z^*(t)) \approx 6.15 \cdot 10^{21}$. Thus, the probability TAI will be achievable at the steady state of our arms race is $P_{\text{TAI}}((F_{\text{max}}(t) \cdot W^*(t))) \approx P_{\text{TAI}}(10^{24} \cdot 6.15 \cdot 10^{21}) \approx 0.89$. This is significantly higher than the corresponding bipolar model, which aligns with our intuition, developed in [Figure 7](#), that the tripolar model leads to significantly more long-term growth than the bipolar model.

4. Conclusion. In this paper, we combine Cotra’s model forecasting the likelihood of developing Transformative AI over a certain time frame with Hamblin-type arms race models. We find that modeling AI development as a bipolar arms race leads to significant short-term

increases in AI spending, but that such models reach equilibrium at relatively low levels of spending. In contrast, modeling AI development as a less stable tripolar arms race leads to a longer period of sustained growth in spending. There are some similarities between the two models, as both predict a sustained period of American dominance over AI spending, fueled by a higher grievance level. However, we also find that a multipolar arms race would significantly accelerate AI development timelines. This would increase the probability that TAI is developed within the next twenty years compared to both the baseline and the bipolar models.

Future work could extend the equilibrium analysis that we begin in [subsection 3.3](#). While we prove some initial results about the long-term behavior of our systems, removing the simplifying assumptions about m_1 and m_2 made in [subsection 3.3](#) could help us better understand the spending dynamics of the two followers in our tripolar model. Studying the rate of convergence of different systems, and how this might depend on our fitted parameters, could also further our understanding of long-term spending dynamics. Additionally, our models remain somewhat sensitive to changes in parameters, especially (for example) when mn is close to 1. Future work could consider the effects of tuning these hyperparameters on the model dynamics. For instance, how might diplomacy (modeled as lowering grievance constants) between nations in an AI arms race impact future spending dynamics? Alternate formulations of this arms race model that are more robust could also be a worthwhile extension of our work.

REFERENCES

- [1] A. COTRA, *Forecasting tai with biological anchors*, July 2020,
<https://docs.google.com/document/d/1IJ6Sr-gPeXdSJugFulwIpvavc0atjHGM82QjIfUSBGQ/edit>.
- [2] J. DING, *Deciphering china's ai dream*, (2018),
https://www.fhi.ox.ac.uk/wp-content/uploads/Deciphering_Chinas_AI-Dream.pdf.
- [3] C. S. GRAY, *The arms race phenomenon*, World Politics, 24 (1971), pp. 39–79,
<http://www.jstor.org/stable/2009706> (accessed 2022-10-10).
- [4] R. L. HAMBLIN, M. HOUT, J. L. L. MILLER, AND B. L. PITCHER, *Arms races: A test of two models*, American Sociological Review, 42 (1977), pp. 338–354, <http://www.jstor.org/stable/2094609> (accessed 2022-10-10).
- [5] J. E. HUNTER, *Mathematical models of a three-nation arms race*, The Journal of Conflict Resolution, 24 (1980), pp. 241–252, <http://www.jstor.org/stable/173852> (accessed 2022-12-13).
- [6] H. KARNOFSKY, *Some background on our views regarding advanced artificial intelligence*, May 2016,
openphilanthropy.org/research/some-background-on-our-views-regarding-advanced-artificial-intelligence.
- [7] D. LI, T. TONG, AND Y. XIAO, *Is china emerging as the global leader in ai?*, (2021),
hbr.org/2021/02/is-china-emerging-as-the-global-leader-in-ai.
- [8] L. RICHARDSON, N. RASHEVSKY, AND E. TRUCCO, *Arms and Insecurity: A Mathematical Study of the Causes and Origins of War*, Boxwood Press, 1960,
<https://books.google.com/books?id=JnjeAAAAMAAJ>.
- [9] *How globalsearch and multistart work*, 2022,
<https://www.mathworks.com/help/gads/how-globalsearch-and-multistart-work.html>.
- [10] D. ZHANG, N. MASLEJ, E. BRYNJOLFSSON, J. ETCEMENDY, T. LYONS, J. MANYIKA, H. NGO, J. C. NIEBLES, M. SELLITTO, E. SAKHAEI, Y. SHOHAM, J. CLARK, AND R. PERRAULT, *The ai index 2022 annual report*, 2022, <https://doi.org/10.48550/ARXIV.2205.03468>,
<https://arxiv.org/abs/2205.03468>.

Appendix A. MATLAB Code.

A.1. Fitting a Bipolar Arms Race Model.

```
%% Data Fitting
```


FORECASTING AI DEVELOPMENT WITH ARMS RACE MODELS

```
china = [0.129 0.448 1.808 3.476 6.009 10.24 12.747 9.823 17.21];
usa = [1.941 4.504 5.831 7.904 11.299 16.13 21.648 26.907 52.872];

numstartpoints = 100;
LowerBounds = [0.561 0.122 0.165 0.452];
UpperBounds = [46.55 2.376 1.396 3.431];
xstart=.5*(LowerBounds+UpperBounds); % initial param values

%% define problem
problem = createOptimProblem('fmincon','objective',@SOLVE_RACE,'x0',xstart,'lb',LowerBounds,'ub',UpperBounds);
problem.options = optimoptions(problem.options,'MaxFunEvals',9999,'MaxIter',9999);

%% run multistart to generate parameters
ms=MultiStart('UseParallel',true,'Display','iter');
[b,fval,exitflag,output,manymins]=run(ms,problem,numstartpoints); %runs the multistart
Parameters = manymins(1).X;

%% Outputs state variables for "best" fit
disp(Parameters);
s = Parameters(1);
n = Parameters(2);
b = Parameters(3);
m = Parameters(4);

%% initial conditions and model dynamics
x0 = 1.941; % leader
```

```
y0 = 0.129; % follower

s = 4.1273;
n = 0.9332;
b = 0.6300;
m = 0.8633;

numTimeSteps = 8;
x = zeros(1,numTimeSteps);
y = zeros(1,numTimeSteps);
x(1) = x0;
y(1) = y0;

for t=1:numTimeSteps
    x(t+1) = s*y(t)^n;
    y(t+1) = b*x(t+1)^m;
    % y(t+1) = b*exp(m*x(t+1));
end

%% Plot the results
plot(x,'.-','MarkerSize',12);
hold on
plot(y,'.-','MarkerSize',12);
hold off
title('Predicted Spending over Time', 'FontSize', 24);
set(gca,'FontSize',18)
```

```
xlabel('Years since 2013')
ylabel('Spending (billions USD)')

figure()
plot(y, '-.', 'MarkerSize', 12);
hold on
plot(china, '-.', 'MarkerSize', 12);
hold off

title('Predicted v Actual CHN', 'FontSize', 24);
set(gca, 'FontSize', 18)
xlabel('Years since 2013')
ylabel('Spending (billions USD)')

figure()
plot(usa, '-.', 'MarkerSize', 12);
hold on
plot(x, '-.', 'MarkerSize', 12);
hold off

title('Predicted v Actual USA', 'FontSize', 24);
set(gca, 'FontSize', 18)
xlabel('Years since 2013')
ylabel('Spending (billions USD)')

% for use by optimizer
function value=SOLVE_RACE(params)
s = params(:,1);
```

```
n = params(:,2);
b = params(:,3);
m = params(:,4);

% data
%x_real = [1.941 4.504 5.831 7.904 11.299 16.13 21.648 26.907];
%y_real = [0.129 0.448 1.808 3.476 6.009 10.24 12.747 9.823];
x_real = [1.941 4.504 5.831 7.904 11.299 16.13 21.648 26.907 52.872];
y_real = [0.129 0.448 1.808 3.476 6.009 10.24 12.747 9.823 17.21];
numTimeSteps = 8;
x = zeros(1,numTimeSteps);
y = zeros(1,numTimeSteps);
x(1) = 1.941;
y(1) = 0.129;
for t=1:numTimeSteps
    x(t+1) = s*y(t)^n;
    y(t+1) = b*x(t+1)^m;
    % y(t+1) = b*exp(m*x(t+1));
end
diff1 = x_real - reshape(x,size(x_real));
diff2 = y_real - reshape(y,size(y_real));
value=norm(diff1,2) + norm(diff2,2);

if value > 999999999
    value = 999999999;
end
```

```
end
```

A.2. Fitting a Tripolar Arms Race Model.

```
%% Data Fitting
```

```
china = [0.129 0.448 1.808 3.476 6.009 10.24 12.747 9.823 17.21];
```

```
usa = [1.941 4.504 5.831 7.904 11.299 16.13 21.648 26.907 52.872];
```

```
eu = [0.471 0.575 0.314 0.531 0.881 1.355 2.233 2.013 6.424];
```

```
numstartpoints = 100;
```

```
% parameters: [s n b1 m1 b2 m2]
```

```
LowerBounds = [0.561 0.122 0.165 0.452 0.165 0.452];
```

```
UpperBounds = [46.55 2.376 1.396 3.431 1.396 3.431];
```

```
xstart=.5*(LowerBounds+UpperBounds); % initial param values
```

```
%% define problem
```

```
problem = createOptimProblem('fmincon','objective',@SOLVE_RACE,'x0',xstart,'lb',LowerBounds,'ul',UpperBounds);
```

```
problem.options = optimoptions(problem.options,'MaxFunEvals',9999,'MaxIter',9999);
```

```
%% run multistart to generate parameters
```

```
ms=MultiStart('UseParallel',true,'Display','iter');
```

```
[b,fval,exitflag,output,manymins]=run(ms,problem,numstartpoints); %runs the multistart
```

```
Parameters = manymins(1).X;
```

```
s = Parameters(1);
```

```
n = Parameters(2);
```

```
b1 = Parameters(3);
```

```
m1 = Parameters(4);
```

```
b2 = Parameters(5);
```

```
m2 = Parameters(6);

%% initial conditions
x0 = 1.941; % leader
y0 = 0.129; % follower
z0 = 0.471; % follower 2

numTimeSteps = 8;
x = zeros(1,numTimeSteps);
y = zeros(1,numTimeSteps);
z = zeros(1,numTimeSteps);
x(1) = x0;
y(1) = y0;
z(1) = z0;

for t=1:numTimeSteps
    x(t+1) = s*(y(t)^n + z(t)^n);
    y(t+1) = b1*x(t+1)^m1;
    z(t+1) = b2*x(t+1)^m2;
end

%% Plot the results
subplot(1,3,1);
plot(y,'.-','MarkerSize',12);
hold on
plot(china,'.-','MarkerSize',12);
```

```
hold off

title('Predicted v Actual CHN', 'FontSize', 24);

set(gca,'FontSize',18)

xlabel('Years since 2013')

ylabel('Spending (billions USD)')


subplot(1,3,2);

plot(usa,'.-','MarkerSize',12);

hold on

plot(x,'.-','MarkerSize',12);

hold off

title('Predicted v Actual USA', 'FontSize', 24);

set(gca,'FontSize',18)

xlabel('Years since 2013')

ylabel('Spending (billions USD)')


subplot(1,3,3);

plot(eu,'.-','MarkerSize',12);

hold on

plot(z,'.-','MarkerSize',12);

hold off

title('Predicted v Actual EU', 'FontSize', 24);

set(gca,'FontSize',18)

xlabel('Years since 2013')

ylabel('Spending (billions USD)')
```

```
%% to be used by multistart

function value=SOLVE_RACE(params)

s = params(:,1);
n = params(:,2);
b1 = params(:,3);
m1 = params(:,4);
b2 = params(:,5);
m2 = params(:,6);

% data

x_real = [1.941 4.504 5.831 7.904 11.299 16.13 21.648 26.907 52.872];
y_real = [0.129 0.448 1.808 3.476 6.009 10.24 12.747 9.823 17.21];
z_real = [0.471 0.575 0.314 0.531 0.881 1.355 2.233 2.013 6.424];


numTimeSteps = 8;
x = zeros(1,numTimeSteps);
y = zeros(1,numTimeSteps);
z = zeros(1,numTimeSteps);
x(1) = 1.941;
y(1) = 0.129;
z(1) = 0.471;

for t=1:numTimeSteps
    x(t+1) = s*(y(t)^n + z(t)^n);
    y(t+1) = b1*x(t+1)^m1;
    z(t+1) = b2*x(t+1)^m2;
end
```



```
diff1 = x_real - reshape(x,size(x_real));
diff2 = y_real - reshape(y,size(y_real));
diff3 = z_real - reshape(z,size(z_real));
value=norm(diff1,2) + norm(diff2,2) + norm(diff3,2);

if value > 999999999
    value = 999999999;
end

end
```

A.3. Generating Bipolar Model Predictions.

```
%% initial conditions and model dynamics
x0 = 52.872; % leader
y0 = 17.21; % follower

s = 4.1273;
n = 0.9332;
b = 1.6300;
m = 0.8633;

numTimeSteps = 20;
x = zeros(1,numTimeSteps);
y = zeros(1,numTimeSteps);
x(1) = x0;
y(1) = y0;
```

```
for t=1:numTimeSteps
    x(t+1) = s*y(t)^n;
    y(t+1) = b*x(t+1)^m;
end

s = zeros(1,numTimeSteps);
for t=1:numTimeSteps+1
    s(t) = x(t) + y(t);
end

%% scaling
s = log10(s);
s = s + 9 - s(1);
writematrix(reshape(s,[21,1]), 's2.csv')

%% Plot the results
plot(x, '.-', 'MarkerSize', 12);
% plot(log10(x)+9, '.-', 'MarkerSize', 12);
hold on
plot(y, '.-', 'MarkerSize', 12);
% plot(log10(y)+9, '.-', 'MarkerSize', 12);
hold off
title('Predicted Spending over Time', 'FontSize', 24);
set(gca, 'FontSize', 18)
xlabel('Years since 2020')
ylabel('Spending (Billions USD)')
```

A.4. Generating Tripolar Model Predictions.

```
%% initial conditions and model dynamics
```

```
x0 = 52.872; % leader
```

```
y0 = 17.21; % follower 1 (china)
```

```
z0 = 6.424; % follower 2 (eu)
```

```
s = 2.433;
```

```
n = 1.0766;
```

```
b1 = 0.7824;
```

```
m1 = 0.7999;
```

```
b2 = 0.165;
```

```
m2 = 1.0137;
```

```
numTimeSteps = 20;
```

```
x = zeros(1,numTimeSteps);
```

```
y = zeros(1,numTimeSteps);
```

```
z = zeros(1,numTimeSteps);
```

```
x(1) = x0;
```

```
y(1) = y0;
```

```
z(1) = z0;
```

```
for t=1:numTimeSteps
```

```
    x(t+1) = s*(y(t)^n + z(t)^n);
```

```
    y(t+1) = b1*x(t+1)^m1;
```

```
    z(t+1) = b2*x(t+1)^m2;
```

```
end
```

```

s = zeros(1,numTimeSteps);
for t=1:numTimeSteps+1
    s(t) = x(t) + y(t);
end
%% scaling
s = log10(s);
s = s + 9 - s(1);
writematrix(reshape(s,[numTimeSteps + 1,1]),'s3_extended.csv')

%% Plot the results
plot(x,'.-','MarkerSize',12);
hold on
plot(y,'.-','MarkerSize',12);
plot(z,'.-','MarkerSize',12);
hold off
title('Predicted Spending over Time', 'FontSize', 24);
set(gca,'FontSize',18)
xlabel('Years since 2020')
ylabel('Spending (Billions USD)')

```

A.5. Generating Plots to Compare the Baseline, Bipolar, and Tripolar Models.

```

% log spending
threewayArmsSpend = [9 9.125311829 9.306713455 9.477832277 9.640084553 9.794699386 9.94274999 10.191811829 10.440924455 10.690037081 10.939149733 11.188262381 11.437375029 11.686487677 11.935600325 12.184712973 12.433825621 12.682938269 12.932050917 13.181163565 13.430276213 13.679388861 13.928501509 14.177614157 14.426726805 14.675839453 14.924952101 15.174064749 15.423177397 15.672290045 15.921402693 16.170515341 16.419627989 16.668740637 16.917853285 17.166965933 17.416078581 17.665191229 17.914303877 18.163416525 18.412529173 18.661641821 18.910754469 19.159867117 19.408979765 19.658092413 19.907205061 20.156317709 20.405430357 20.654543005 20.903655653 21.152768301 21.401880949 21.650993597 21.900106245 22.149218893 22.398331541 22.647444189 22.896556837 23.145669485 23.394782133 23.643894781 23.893007429 24.142120077 24.391232725 24.640345373 24.889458021 25.138570669 25.387683317 25.636795965 25.885908613 26.135021261 26.384133909 26.633246557 26.882359205 27.131471853 27.380584501 27.629697149 27.878809797 28.127922445 28.377035093 28.626147741 28.875260389 29.124373037 29.373485685 29.622598333 29.871710981 30.120823629 30.369936277 30.619048925 30.868161573 31.117274221 31.366386869 31.615499517 31.864612165 32.113724813 32.362837461 32.611950109 32.861062757 33.110175405 33.359288053 33.608400701 33.857513349 34.106625997 34.355738645 34.604851293 34.853963941 35.103076589 35.352189237 35.601301885 35.850414533 36.099527181 36.348639829 36.597752477 36.846865125 37.095977773 37.345090421 37.594203069 37.843315717 38.092428365 38.341541013 38.590653661 38.839766309 39.088878957 39.337991605 39.587104253 39.836216901 40.085329549 40.334442197 40.583554845 40.832667493 41.081780141 41.330892789 41.580005437 41.829118085 42.078230733 42.327343381 42.576456029 42.825568677 43.074681325 43.323793973 43.572906621 43.822019269 44.071131917 44.320244565 44.569357213 44.818469861 45.067582509 45.316695157 45.565807805 45.814920453 46.064033101 46.313145749 46.562258397 46.811371045 47.060483693 47.309596341 47.558708989 47.807821637 48.056934285 48.306046933 48.555159581 48.804272229 49.053384877 49.302497525 49.551610173 49.800722821 50.049835469 50.298948117 50.548060765 50.797173413 51.046286061 51.295398709 51.544511357 51.793624005 52.042736653 52.291849301 52.540961949 52.790074597 53.039187245 53.288299893 53.537412541 53.786525189 54.035637837 54.284750485 54.533863133 54.782975781 55.032088429 55.281201077 55.530313725 55.779426373 56.028539021 56.277651669 56.526764317 56.775876965 57.024989613 57.274102261 57.523214909 57.772327557 58.021440205 58.270552853 58.519665501 58.768778149 59.017890797 59.267003445 59.516116093 59.765228741 60.014341389 60.263454037 60.512566685 60.761679333 61.010791981 61.259904629 61.509017277 61.758129925 62.007242573 62.256355221 62.505467869 62.754580517 63.003693165 63.252805813 63.501918461 63.751031109 64.000143757 64.249256405 64.498369053 64.747481701 64.996594349 65.245706997 65.494819645 65.743932293 65.993044941 66.242157589 66.491270237 66.740382885 66.989495533 67.238608181 67.487720829 67.736833477 67.985946125 68.235058773 68.484171421 68.733284069 68.982396717 69.231509365 69.480622013 69.729734661 69.978847309 70.227959957 70.477072605 70.726185253 70.975297901 71.224410549 71.473523197 71.722635845 71.971748493 72.220861141 72.469973789 72.719086437 72.968199085 73.217311733 73.466424381 73.715537029 73.964649677 74.213762325 74.462874973 74.711987621 74.961099269 75.210211917 75.459324565 75.708437213 75.957549861 76.206662509 76.455775157 76.704887805 76.953999453 77.203112101 77.452224749 77.701337397 77.950449045 78.199561693 78.448674341 78.697786989 78.946899637 79.196012285 79.445124933 79.694237581 79.943350229 80.192462877 80.441575525 80.690688173 80.939799821 81.188912469 81.438025117 81.687137765 81.936250413 82.185363061 82.434475709 82.683588357 82.932699905 83.181812553 83.430925201 83.680037849 83.929150497 84.178263145 84.427375793 84.676488441 84.925601089 85.174713737 85.423826385 85.672939033 85.922051681 86.171164329 86.420276977 86.669389625 86.918502273 87.167614921 87.416727569 87.665840217 87.914952865 88.164065513 88.413178161 88.662290809 88.911403457 89.160516105 89.409628753 89.658741401 89.907854049 90.156966697 90.406079345 90.655191993 90.904304641 91.153417289 91.402529937 91.651642585 91.900755233 92.149867881 92.398980529 92.648093177 92.897205825 93.146318473 93.395431121 93.644543769 93.893656417 94.142769065 94.391881713 94.640994361 94.890107009 95.139219657 95.388332305 95.637444953 95.886557601 96.135670249 96.384782897 96.633895545 96.883008193 97.132120841 97.381233489 97.630346137 97.879458785 98.128571433 98.377684081 98.626796729 98.875909377 99.125022025 99.374134673 99.623247321 99.872359969 100.121472617 100.370585265 100.619697913 100.868810561 101.117923209 101.367035857 101.616148505 101.865261153 102.114373801 102.363486449 102.612599097 102.861711745 103.110824393 103.359937041 103.609049689 103.858162337 104.107274985 104.356387633 104.605499281 104.854611929 105.103724577 105.352837225 105.601949873 105.851062521 106.100175169 106.349287817 106.598400465 106.847513113 107.096625761 107.345738409 107.594851057 107.843963705 108.093076353 108.342188901 108.591301549 108.840414197 109.089526845 109.338639493 109.587752141 109.836864789 110.085977437 110.335090085 110.584202733 110.833315381 111.082428029 111.331540677 111.580653325 111.829765973 112.078878621 112.327991269 112.577103917 112.826216565 113.075329213 113.324441861 113.573554509 113.822667157 114.071779805 114.320892453 114.569999999 114.819112647 115.068225295 115.317337943 115.566450591 115.815563239 116.064675887 116.313788535 116.562901183 116.812013831 117.061126479 117.310239127 117.559351775 117.808464423 118.057577071 118.306689719 118.555802367 118.804915015 119.054027663 119.303140311 119.552252959 119.801365607 120.050478255 120.299590903 120.548703551 120.797816199 121.046928847 121.296041495 121.545154143 121.794266791 122.043379439 122.292492087 122.541604735 122.790717383 123.039830031 123.288942679 123.538055327 123.787167975 124.036280623 124.285393271 124.534505919 124.783618567 125.032731215 125.281843863 125.530956511 125.780069159 126.029181807 126.278294455 126.527407103 126.776519751 127.025632399 127.274745047 127.523857695 127.772970343 128.022082991 128.271195639 128.520308287 128.769420935 129.018533583 129.267646231 129.516758879 129.765871527 130.014984175 130.264096823 130.513209471 130.762322119 131.011434767 131.260547415 131.509660063 131.758772711 132.007885359 132.256998007 132.506110655 132.755223303 133.004335951 133.253448599 133.502561247 133.751673895 134.000786543 134.249899191 134.499011839 134.748124487 135.000000000 135.250000000 135.500000000 135.750000000 136.000000000 136.250000000 136.500000000 136.750000000 137.000000000 137.250000000 137.500000000 137.750000000 138.000000000 138.250000000 138.500000000 138.750000000 139.000000000 139.250000000 139.500000000 139.750000000 140.000000000 140.250000000 140.500000000 140.750000000 141.000000000 141.250000000 141.500000000 141.750000000 142.000000000 142.250000000 142.500000000 142.750000000 143.000000000 143.250000000 143.500000000 143.750000000 144.000000000 144.250000000 144.500000000 144.750000000 145.000000000 145.250000000 145.500000000 145.750000000 146.000000000 146.250000000 146.500000000 146.750000000 147.000000000 147.250000000 147.500000000 147.750000000 148.000000000 148.250000000 148.500000000 148.750000000 149.000000000 149.250000000 149.500000000 149.750000000 150.000000000 150.250000000 150.500000000 150.750000000 151.000000000 151.250000000 151.500000000 151.750000000 152.000000000 152.250000000 152.500000000 152.750000000 153.000000000 153.250000000 153.500000000 153.750000000 154.000000000 154.250000000 154.500000000 154.750000000 155.000000000 155.250000000 155.500000000 155.750000000 156.000000000 156.250000000 156.500000000 156.750000000 157.000000000 157.250000000 157.500000000 157.750000000 158.000000000 158.250000000 158.500000000 158.750000000 159.000000000 159.250000000 159.500000000 159.750000000 160.000000000 160.250000000 160.500000000 160.750000000 161.000000000 161.250000000 161.500000000 161.750000000 162.000000000 162.250000000 162.500000000 162.750000000 163.000000000 163.250000000 163.500000000 163.750000000 164.000000000 164.250000000 164.500000000 164.750000000 165.000000000 165.250000000 165.500000000 165.750000000 166.000000000 166.250000000 166.500000000 166.750000000 167.000000000 167.250000000 167.500000000 167.750000000 168.000000000 168.250000000 168.500000000 168.750000000 169.000000000 169.250000000 169.500000000 169.750000000 170.000000000 170.250000000 170.500000000 170.750000000 171.000000000 171.250000000 171.500000000 171.750000000 172.000000000 172.250000000 172.500000000 172.750000000 173.000000000 173.250000000 173.500000000 173.750000000 174.000000000 174.250000000 174.500000000 174.750000000 175.000000000 175.250000000 175.500000000 175.750000000 176.000000000 176.250000000 176.500000000 176.750000000 177.000000000 177.250000000 177.500000000 177.750000000 178.000000000 178.250000000 178.500000000 178.750000000 179.000000000 179.250000000 179.500000000 179.750000000 180.000000000 180.250000000 180.500000000 180.750000000 181.000000000 181.250000000 181.500000000 181.750000000 182.000000000 182.250000000 182.500000000 182.750000000 183.000000000 183.250000000 183.500000000 183.750000000 184.000000000 184.250000000 184.500000000 184.750000000 185.000000000 185.250000000 185.500000000 185.750000000 186.000000000 186.250000000 186.500000000 186.750000000 187.000000000 187.250000000 187.500000000 187.750000000 188.000000000 188.250000000 188.500000000 188.750000000 189.000000000 189.250000000 189.500000000 189.750000000 190.000000000 190.250000000 190.500000000 190.750000000 191.000000000 191.250000000 191.500000000 191.750000000 192.000000000 192.250000000 192.500000000 192.750000000 193.000000000 193.250000000 193.500000000 193.750000000 194.000000000 194.250000000 194.500000000 194.750000000 195.000000000 195.250000000 195.500000000 195.750000000 196.000000000 196.250000000 196.500000000 196.750000000 197.000000000 197.250000000 197.500000000 197.750000000 198.000000000 198.250000000 198.500000000 198.750000000 199.000000000 199.250000000 199.500000000 199.750000000 200.000000000 200.250000000 200.500000000 200.750000000 201.000000000 201.250000000 201.500000000 201.750000000 202.000000000 202.250000000 202.500000000 202.750000000 203.000000000 203.250000000 203.500000000 203.750000000 204.000000000 204.250000000 204.500000000 204.750000000 205.000000000 205.250000000 205.500000000 205.750000000 206.000000000 206.250000000 206.500000000 206.750000000 207.000000000 207.250000000 207.500000000 207.750000000 208.000000000 208.250000000 208.500000000 208.750000000 209.000000000 209.250000000 209.500000000 209.750000000 210.000000000 210.250000000 210.500000000 210.750000000 211.000000000 211.250000000 211.500000000 211.750000000 212.000000000 212.250000000 212.500000000 212.750000000 213.000000000 213.250000000 213.500000000 213.750000000 214.000000000 214.250000000 214.500000000 214.750000000 215.000000000 215.250000000 215.500000000 215.750000000 216.000000000 216.250000000 216.500000000 216.750000000 217.000000000 217.250000000 217.500000000 217.750000000 218.000000000 218.250000000 218.500000000 218.750000000 219.000000000 219.250000000 219.500000000 219.750000000 220.000000000 220.250000000 220.500000000 220.750000000 221.000000000 221.250000000 221.500000000 221.750000000 222.000000000 222.250000000 222.500000000 222.750000000 223.000000000 223.250000000 223.500000000 223.750000000 224.000000000 224.250000000 224.500000000 224.750000000 225.000000000 225.250000000 225.500000000 225.750000000 226.000000000 226.250000000 226.500000000 226.750000000 227.000000000 227.250000000 227.500000000 227.750000000 228.000000000 228.250000000 228.500000000 228.750000000 229.000000000 229.250000000 229.5000000
```

FORECASTING AI DEVELOPMENT WITH ARMS RACE MODELS

```
armsProb = [2.62 3.56 5.44 7.55 9.68 11.79 13.88 15.90 17.91 19.91 21.84 23.78 25.71 27.63 29.1]
normProb = [2.62 3.34 4.22 5.28 6.47 7.83 9.27 10.85 12.54 14.32 16.21 18.20 20.28 22.45 24.66]

%% Plot the results

plot(normSpend, '-.', 'MarkerSize', 12);

hold on

plot(armsSpend, '-.', 'MarkerSize', 12);

plot(threewayArmsSpend, '-.', 'MarkerSize', 12);

hold off

title('Predicted Spending over Time', 'FontSize', 24);

set(gca, 'FontSize', 18)

xlabel('Years since 2020')

ylabel('Log of Spending (USD)')

figure()

plot(normProb, '-.', 'MarkerSize', 12);

hold on

plot(armsProb, '-.', 'MarkerSize', 12);

plot(threewayArmsProb, '-.', 'MarkerSize', 12);

hold off

title('Probability of Having Enough Compute for TAI', 'FontSize', 24);

set(gca, 'FontSize', 18)

xlabel('Years since 2020')

ylabel('% Chance of Having Enough Compute for TAI')
```