Regulating Artificial Intelligence[*]

Joao Guerreiro[†] Sergio Rebelo[‡] Pedro Teles[§]

December 2023

Abstract

We consider an environment in which there is substantial uncertainty about the potential adverse external effects of AI algorithms. We find that subjecting algorithm implementation to regulatory approval or mandating testing is insufficient to implement the social optimum. When testing costs are low, a combination of mandatory testing for external effects and making developers liable for the adverse external effects of their algorithms comes close to implementing the social optimum even when developers have limited liability.

J.E.L. Classification: H21, O33
Keywords: AI, regulation

[*] We thank Alessandro Pavan for his comments and Ramya Raghavan for excellent research assistance.

[†] UCLA.

[‡] Northwestern University, NBER, and CEPR.

[§] Banco de Portugal, Catolica-Lisbon School of Business & Economics, and CEPR.

1 Introduction

In 1950, Isaac Asimov published *I, Robot*, a collection of short stories about the dilemmas of a world where robots powered by artificial intelligence (AI) interact with humans. Recent advances in AI have brought these dilemmas from the realm of science fiction to the pages of newspapers and the halls of parliaments. In this paper, we discuss the efficacy of different approaches to AI regulation.

Over the past decade, the declining costs of computing power and the availability of vast data sets allowed neural networks and other forms of AI to accomplish remarkable feats. Reinforcement learning algorithms beat humans at games of perfect information, like chess and go (Silver et al., 2017, Silver et al., 2018). AI algorithms outperformed humans in games of imperfect information, such as poker (Brown and Sandholm, 2019). Convolutional neural networks achieved remarkable accuracy in image recognition tasks (Langlotz et al., 2019). Natural language processing models like Generative Pre-trained Transformers have made significant strides in language understanding, translation, and content generation (Eloundou, Manning, Mishkin, and Rock, 2023). AI algorithms, in general, have improved prediction accuracy in many domains relevant to business applications (Agrawal, Gans, and Goldfarb, 2022).

These and other breakthroughs in AI hold the promise of delivering significant benefits to society. However, they also carry the risk of imposing considerable societal costs. These costs include negative externalities, such as fueling political polarization, facilitating fraud, disseminating false information, jeopardizing financial stability, and weakening democracies (Beraja, Kao, Yang, and Yuchtman, 2023). Other costs take the form of "internalities," a term coined by Herrnstein, Loewenstein, Prelec, and Vaughan Jr (1993) that refers to situations in which individuals are manipulated to act against their self-interest through misinformation or exploitation of self-control and time inconsistency problems.

In May 2023, a consortium of prominent figures in the field of AI signed a statement declaring that “Addressing the existential risks posed by AI should be a global priority, on par with other worldwide challenges like pandemics and nuclear warfare.” The leaders of the G7 nations initiated the Hiroshima AI Process to harmonize AI regulation.

Europe and the United States have started to design regulatory frameworks to address the challenges posed by AI (see European Commission, 2020, Benifei and Tudorache, 2023, and Biden, 2023). Two approaches proposed so far include mandatory testing of AI algorithms and holding AI developers accountable for the adverse outcomes resulting from using their technology. Policymakers are also considering classifying AI technologies into risk tiers (unacceptable, high, limited, and minimal risk), forbidding the development of algorithms that create unacceptable risks (European Commission, 2022).

We assess these approaches to regulate AI using a model designed to capture the key aspect of ongoing developments in AI: there is substantial uncertainty about the resulting societal costs and benefits. We evaluate the efficacy of different regulatory frameworks by comparing the resulting equilibrium outcomes with the social optimum, i.e. the solution to the planner’s problem.

As discussed earlier, the social costs of AI fall into two broad categories: negative externalities and internalities. In the main text, we focus our discussion on the potential negative externalities generated by AI algorithms. In the Appendix, we revisit our results for scenarios where AI algorithms can cause internalities.

We explore two settings. In both settings, an AI developer chooses the novelty of its AI algorithm relative to the state of the art. There is ex-ante uncertainty about the negative externalities that this algorithm might cause. This uncertainty grows with the difference between the new algorithm’s novelty and the status quo.

In the first setting, uncertainty is not resolved until an AI algorithm is fully implemented, and this implementation is irreversible. Potential negative externalities drive a wedge between the social optimum and the unregulated equilibrium. The planner wants to be more cautious than private markets–the optimal level of AI novelty for society is lower than what naturally emerges in an unregulated setting.

In the second setting, uncertainty regarding potential negative externalities can be resolved through experimentation, which we call beta testing. This testing involves making the algorithm available to a small group of households and using the test results to decide whether to make the algorithm available to the population as a whole. Developers regularly engage in beta testing to assess how effective their algorithm is from a user's perspective. The beta testing we emphasize in this paper serves a distinct purpose: to measure an algorithm's social costs (negative externalities or internalities).

In the unregulated equilibrium, the developer has weaker incentives for beta testing than the planner. Once again, the planner exhibits greater caution than private markets.

We show three results. First, subjecting algorithm release to regulatory approval is insufficient to implement the social optimum–developers still have an incentive to create algorithms that are too risky. Second, simply holding developers accountable for the negative externalities produced by their algorithms implements the social optimum if developers are not protected by limited liability. However, this policy is also insufficient to implement the social optimum if developers are protected by limited liability. Third, we can achieve a solution close to the social optimum when regulators mandate beta testing to assess externalities and hold developers liable for the adverse externalities of their algorithms, even if there is limited liability. One advantage of this solution is that developers do not need to seek regulatory approval before implementing their algorithms.

In Appendix A, we show that the impact of negative externalities and internalities is broadly similar. One interesting difference is that developers refrain from releasing algorithms with large externalities but do not refrain from releasing algorithms with large internalities.

Our paper is related to four important strands of literature. The first studies the value of experimentation (e.g., Callander, 2011 and Ilut and Valchev, 2023). The second analyses settings relevant to the design and execution of clinical trials. These situations feature multiple options and unknown rewards, commonly known as the multi-armed bandit problem (e.g., Thompson, 1933 and Gittins, 1974). The third considers the importance of data as an input into AI algorithms (e.g., Jones and Tonetti, 2020 and Farboodi and Veldkamp, 2021). The fourth researches the impact of AI on the economy (e.g., Burstein, Morales, and Vogel, 2019, Acemoglu and Restrepo, 2022, and Jones, 2023).

In Section 2, we study the model without beta-testing. We introduce beta-testing in Section 3. In Section 4, we evaluate different regulatory proposals. We study scenarios in which AI algorithms create internalities in Section A. Section 5 concludes.

2 Model without beta testing

This section considers a model in which an AI algorithm cannot be tested before it is released and in which the release is irreversible. The model features a continuum of identical households and a single AI developer. We discuss the household problem, the problem of the AI developer, and the unregulated equilibrium. Then, we characterize the social optimum and compare it with the unregulated equilibrium.

2.1 Unregulated equilibrium

Household problem The economy has a continuum of households indexed by $i \in [0, N]$, where N denotes the number of households in the population. Each household has a constant exogenous income level denoted by y. Households decide whether to purchase a license to use an AI algorithm with novelty ℓ at a price p.

Their utility, $\mathcal{U}_i$, has a quasi-linear form:

$$\mathcal{U}_i = y + [u(\ell)\,\mu - p] \times \mathcal{I}_i - \mathbb{E}[e^2]. \tag{1}$$

The utility derived from using the AI algorithm is $u(\ell)\,\mu$. We assume that the function u is increasing ($u' > 0$) and concave ($u'' < 0$) in the novelty level, ℓ, and satisfies the Inada condition ($\lim_{\ell \downarrow 0} u'(\ell) = \infty$). We also normalize $u(0) = 0$, where $\ell = 0$ represents the status quo. The indicator function $\mathcal{I}_i$ takes the value one if household i buys the AI license and zero otherwise. To capture positive network externalities, we assume that this utility is proportional to the number of users $\mu = \int \mathcal{I}_i di$.

AI usage can cause a negative externality e that reduces utility by e^2. We assume that the externality is proportional to the measure of users, μ, and takes the form:

$$e = \phi(\ell) \times \mu,$$

For each value of the novelty level ℓ, $\phi(\ell)$ is a random variable. Both positive and negative values of $\phi(\ell)$ represent undesirable, negative externalities.

We assume that the distribution $\phi(\ell)$ satisfies two properties. First, the expected value of $\phi(\ell)$ is zero:

$$\mathbb{E}[\phi(\ell)] = 0.$$

Second, the uncertainty about the potential externality is an increasing function of the novelty level ℓ. Let $\sigma^2(\ell)$ denote the uncertainty about the potential externality of an algorithm with novelty level ℓ:

$$\sigma^2(\ell) = \mathbb{E}\left[\phi(\ell)^2\right].$$

We assume that $\sigma^2(\ell)$ is increasing and convex in ℓ, and $\sigma(0) = 0$, i.e., there is no uncertainty in the status quo.

Replacing e in equation (1), we can write the household problem as choosing $\mathcal{I}_i$ to maximize

$$\mathcal{U}_i = y + [u(\ell)\mu - p] \times \mathcal{I}_i - \sigma^2(\ell)\mu^2.$$

Households are willing to purchase a license to use the AI algorithm whenever private benefits exceed the price,

$$u\left(\ell\right)\mu \geq p.$$

AI developer's problem We consider a single AI developer who chooses the algorithm's novelty level, ℓ, the license price, p, and the number of available licenses, μ.

The cost of developing an algorithm with novelty level ℓ is $f(\ell)$. This cost is increasing and convex in ℓ and $f(0) = 0$.

The developer experiences disutility from the externality in the same way that households do. However, the developer does not take into account the negative externality endured by the households. The expected utility of the developer depends on the price it chooses as follows:

$$\mathcal{V} = \begin{cases} \mu p - \sigma^2\left(\ell\right)\mu^2 - f\left(\ell\right) & \text{if } p \leq u\left(\ell\right)\mu, \\ 0 - f\left(\ell\right) & \text{if } p > u\left(\ell\right)\mu. \end{cases}$$

If the developer markets the AI algorithm, the optimal license price is $p = u\left(\ell\right)\mu$. The developer uses its monopoly position to capture the entire consumer surplus. This pricing strategy does not generate deadweight losses; it simply redistributes resources from the households to the monopolists.

Given that $p = u\left(\ell\right)\mu$, the optimal levels of ℓ and μ solve the following problem:

$$\max_{\ell\geq 0,\ 0\leq\mu\leq N} u\left(\ell\right)\mu^2 - f\left(\ell\right) - \sigma^2\left(\ell\right)\mu^2.$$

We characterize the solution to this problem in two steps. First, taking ℓ as given, we ask what is the optimal release strategy: how many licenses, μ, should be made available at a price, p. Second, we consider the optimal choice of ℓ from the developer's standpoint.

Given ℓ, the optimal release strategy depends on the sign of $u(\ell) - \sigma^2(\ell)$. If this expression is positive, it is optimal to make the algorithm available to the whole population ($\mu = N$). Otherwise, the algorithm is not released ($\mu = 0$).

Given ℓ, the utility of the AI developer is:

$$\mathcal{V}(\ell) = \begin{cases} \{u(\ell) - \sigma^2(\ell)\} N^2 - f(\ell) & \text{if } u(\ell) - \sigma^2(\ell) \geq 0, \\ -f(\ell) & \text{if } u(\ell) - \sigma^2(\ell) < 0. \end{cases}$$

If $[u(\ell) - \sigma^2(\ell)]N^2 < f(\ell)$ for all ℓ, the developer produces no algorithm.

When it is optimal to produce an algorithm and the solution for ℓ is interior, it satisfies the first-order condition:

$$\left\{u'(\ell) - \frac{\partial \sigma^2(\ell)}{\partial \ell}\right\} N^2 - f'(\ell) = 0.$$

Unregulated equilibrium We now describe the characteristics of an equilibrium without regulation in which ℓ has an interior value. The superscript e denotes the values of various variables in this equilibrium. These variables satisfy the following conditions:

$$\mu^e = N,$$
$$\left\{u'(\ell^e) - \frac{\partial \sigma^2(\ell^e)}{\partial \ell}\right\} N^2 - f'(\ell^e) = 0,$$

and ℓ^e is such that the utility of the developer is positive,

$$\mathcal{V}(\ell^e) = \left\{u(\ell^e) - \sigma^2(\ell^e)\right\} N^2 - f(\ell^e) \geq 0.$$

2.2 The planner's problem

The total household welfare in an economy in which μ households use the AI algorithm is

$$\int_0^N \mathcal{U}_i di = Ny + \{u(\ell)\mu - p\}\mu - N\sigma^2(\ell)\mu^2.$$

Social welfare is the sum of the households' and developer's utilities:

$$\mathcal{W} = \int_0^N \mathcal{U}_i di + \mathcal{V} = Ny + \left\{ u\left(\ell\right) - \left(N+1\right)\sigma^2\left(\ell\right) \right\} \mu^2 - f\left(\ell\right).$$

With quasi-linear utility, we can think of this social welfare function as maximizing the total surplus in the economy.

The following proposition compares the social optimum with the unregulated equilibrium.

Proposition 1 (Conservatism in ℓ)**.** *The socially desirable novelty level, ℓ^*, is lower than the level that emerges in the unregulated equilibrium, ℓ^e.*

In the appendix, we prove this proposition using monotone comparative statics. Below, we sketch a proof for the case in which the solution is interior. The socially optimal solution, ℓ^*, satisfies the first-order condition:

$$\left[u'\left(\ell^*\right) - \left(N+1\right) \frac{\partial \sigma^2\left(\ell^*\right)}{\partial \ell} \right] N^2 - f'\left(\ell^*\right) = 0.$$

The optimality condition for the developer evaluated at the socially optimal level of novelty, ℓ^*, is

$$\left[u'\left(\ell^*\right) - \frac{\partial \sigma^2\left(\ell^*\right)}{\partial \ell} \right] N^2 - f'\left(\ell^*\right) > 0,$$

and therefore

$$\ell^e > \ell^*.$$

The key driver of this result is that when selecting ℓ, the developer disregards the externality on the rest of society.

Regulating AI

One regulatory approach to align the decisions of AI developers with societal interests is to impose an upper bound on the degree of novelty, ℓ, that developers can implement. This method resembles the European Commission's proposal of classifying AI algorithms into risk tiers (unacceptable, high, limited, and minimal) and forbidding the development of algorithms with unacceptable risks (European Commission, 2022).

In the model we have been considering, where there is no beta testing, setting the upper bound for ℓ equal to the socially optimal novelty level is sufficient to implement the first best. As shown in the next section, this result no longer holds in the model with beta testing because incentives to test and implement algorithms are not aligned by simply placing an upper bound on ℓ.

Another regulatory approach is to hold AI developers liable for the social costs of the AI algorithm. As discussed in Section 4, without liability limits, this regulation is sufficient to implement the social optimum in settings with and without beta testing. However, with limited liability, the policy is no longer sufficient to align incentives because AI developers do not fully internalize the external consequences of the AI algorithm when $\phi(\ell)$ is very large. In Section 4, we discuss a combination of beta testing controlled by the regulator and limited liability that approximately implements the social optimum.

3 Model with beta testing

This section considers a two-period version of the previous model. Time is discrete and indexed by $t = 1, 2$. To evaluate the externalities, the developer can test the algorithm in the first period in a sample of μ_1 users. Based on the outcomes of this test, they can then decide whether to release the algorithm in the second period. For simplicity, we consider the case in which the external effects, $\phi(\ell)$, are perfectly revealed after the AI algorithm is tested in the first period.

People assign weights $1-\beta$ and β to the utility of the first and second period, respectively. The model without beta testing is a particular case of this more general model where $\beta = 0$.

As β converges to one, the weight of the first period's utility falls, and the cost of testing in the first period, which is the opportunity cost of not deploying the algorithm in this period, becomes negligible.

As in the previous section, we begin by describing the unregulated equilibrium. We then compute the social optimum and compare it to the unregulated equilibrium.

3.1 Unregulated equilibrium

Household problem The household lives for two periods. They choose whether or not to buy an algorithm license in periods one and two to maximize their lifetime utility given by

$$\mathcal{U}_i = (1-\beta)\left\{y + [u(\ell)\mu_1 - p_1]\mathcal{I}_{1,i} - \mathbb{E}(e_1^2)\right\} + \beta\mathbb{E}\left\{y + [u(\ell)\mu_2 - p_2]\mathcal{I}_{2,i} - e_2^2\right\}.$$

The household is willing to purchase an AI license in period t if the private benefits exceed the price

$$u(\ell)\mu_t \geq p_t.$$

AI developer's problem In period one, the AI developer makes three decisions: which novelty level to develop (ℓ), how many AI licenses to offer for sale (μ_1), and what price to charge for each license (p_1). Development costs are incurred in the first period.

If μ_1 equals zero, the developer obtains no information about the externalities of the AI algorithm in period two and the decision-making process in this period resembles that of the model without beta testing.

If μ_1 is greater than zero, the developer obtains information about the external effects of the AI algorithm in period two. Using this information, the developer chooses μ_2, the number of AI licenses to offer for sale in period two, and p_2, the price per license.

The utility of the developer in period two is,

$$\mathcal{V}_2 = \begin{cases} \mu_2 p_2 - \phi(\ell)^2 \mu_2^2, & \text{if } p_2 \leq u(\ell)\mu_2 \quad \text{and} \quad \mu_1 > 0, \\ \mu_2 p_2 - \sigma^2(\ell)\mu_2^2, & \text{if } p_2 \leq u(\ell)\mu_2 \quad \text{and} \quad \mu_1 = 0, \\ 0, & \text{if } p_2 > u(\ell)\mu_2. \end{cases}$$

As before, the price that maximizes the developer's utility is $p_2 = u(\ell)\mu_2$.

If $\mu_1 > 0$, then $\mu_2 = N$ if $u(\ell) - \phi(\ell)^2 \geq 0$ and $\mu_2 = 0$ otherwise. If $\mu_1 = 0$, then $\mu_2 = N$ if $u(\ell) - \sigma^2(\ell) \geq 0$ and $\mu_2 = 0$ otherwise.

Given that an algorithm with novelty ℓ was developed and μ_1 licenses were sold in the first period, the optimized developer utility in period two, $\mathcal{V}_2^*(\ell, \mu_1)$ is,

$$\mathcal{V}_2^*(\ell, \mu_1) = \max \left\{ u(\ell) - \phi(\ell)^2 \mathcal{I}(\mu_1 > 0) - \sigma^2(\ell)(1 - \mathcal{I}(\mu_1 > 0)), 0 \right\} N^2,$$

where $\mathcal{I}(\mu_1 > 0) = 1$ if $\mu_1 > 0$ and zero otherwise. The * indicates that the value function has been maximized with respect to the choice of price and implementation in period two.

To make the problem interesting, we assume that $\phi(\ell)$ is such that there is a strictly positive probabilities that both $u(\ell) - \phi(\ell)^2 > 0$ and $u(\ell) - \phi(\ell)^2 < 0$. This assumption means that the probability that the AI algorithm is implemented in period two, given the information obtained in period one, is strictly positive but less than one.

Lemma 1 (Private benefits of beta testing in period one). *The developer's expected utility in the second period is higher when there is beta testing in the first period,*

$$\mathbb{E}[\mathcal{V}_2^*(\ell, \mu_1)] > \mathcal{V}_2^*(\ell, 0), \quad \textit{if} \quad \mu_1 > 0.$$

Proof. Let μ_2^0 denote the optimal choice of μ_2 when $\mu_1 = 0$. Note that μ_2^0 is necessarily non-state contingent, so $\mu_2^0 = N$ or $\mu_2^0 = 0$ depending on the degree of uncertainty regarding the externality. Then, if $\mu_1 > 0$:

$$\begin{aligned}\mathbb{E}[\mathcal{V}_2^*(\ell,\mu_1)] &= \mathbb{E}\left\{\max\left[\left(u(\ell)-\phi(\ell)^2\right)N^2, 0\right]\right\} \\ &> \mathbb{E}\left[\left(u(\ell)-\phi(\ell)^2\right)(\mu_2^0)^2\right] = \left(u(\ell)-\sigma^2(\ell)\right)(\mu_2^0)^2 = \mathcal{V}_2^*(\ell,0),\end{aligned}$$

where the strict inequality follows from the fact that the probability that the AI algorithm is implemented in period two, given the information obtained in period one, is strictly positive but less than one. □

The problem in period one is to choose ℓ, μ_1 and p_1 to maximize

$$\mathcal{V} = (1-\beta)\left(\begin{cases}\mu_1 p_1 - \sigma^2(\ell)\mu_1^2, & \text{if } p_1 \leq u(\ell)\mu_1 \\ 0, & \text{if } p_1 > u(\ell)\mu_1\end{cases}\right) + \beta\mathbb{E}[\mathcal{V}_2^*(\ell,\mu_1)] - f(\ell).$$

The optimal price for the developer is $p = u(\ell)\mu_1$.

From the standpoint of period one, it is still optimal to set $\mu_1 = N$ if $u(\ell) - \sigma^2(\ell) \geq 0$ and $\mu_1 = 0$ if $u(\ell) - \sigma^2(\ell) < 0$. However, experimenting in the first period, $\mu_1 > 0$, creates value by generating information that the developer can use in the second period.

Given the discontinuity in information generation from $\mu_1 = 0$ to $\mu_1 > 0$, the problem may have a supremum but not a maximum. For a given ℓ, if $u(\ell) - \sigma^2(\ell) < 0$ then the static optimal decision would be $\mu_1 = 0$. However, choosing an infinitesimal, positive value of μ_1 yields strictly larger utility than setting μ_1 to zero (see Lemma 1). Therefore, the optimal number of households trying the technology in period one should be strictly positive but kept as low as possible ($\mu_1 \downarrow 0$). We refer to this setting as the *experimentation solution*: the developer sells AI licenses to an infinitesimal fraction of households to test the algorithm and then decides whether to sell the algorithm given the information revealed in period two.[1]

[1] We could extend the model to the case in which the information revealed is an increasing function

Proposition 2 (Uncertainty, beta testing, and algorithm release)**.** *In an unregulated equilibrium, the number of user licenses (μ_1^e) offered by the developer in the first period depends on the level of uncertainty as follows:*

1. *The developer does beta testing ($\mu_1^e \downarrow 0$) when the degree of uncertainty is high*

$$\sigma^2(\ell) > u(\ell).$$

2. *The developer foregoes beta testing and releases the AI algorithm to the entire population in the first period ($\mu_1^e = N$) when uncertainty is low*

$$\sigma^2(\ell) \leq u(\ell).$$

In both scenarios, the developer learns the externality of the AI algorithm in the second period and then:

1. *Withdraws the algorithm from the market ($\mu_2^e = 0$) if the personal cost to the developer arising from externalities is substantial,*

$$\phi(\ell)^2 > u(\ell).$$

2. *Makes the algorithm available to the whole population ($\mu_2^e = N$) if the personal cost to the developer arising from externalities is relatively minor*

$$\phi(\ell)^2 \leq u(\ell).$$

Since μ_1^e is always positive, then

$$\mathcal{V}_2^*(\ell, \mu_1^e) = \max\{[u(\ell) - \phi(\ell)^2]N^2, 0\},$$

and so

$$\mathcal{V} = (1-\beta)\max\{[u(\ell) - \sigma^2(\ell)]N^2, 0\} + \beta\mathbb{E}\left[\mathcal{V}_2^*(\ell, \mu_1^e)\right] - f(\ell).$$

of the number of households involved in beta testing. In this extension, μ_1 would still be positive, but the model is more complex.

3.2 The planner's problem

We consider a central planner who can decide, in the first period, both the novelty of the AI algorithm developed and the number of households that can access it. If the AI algorithm is implemented in the first period, the planner learns its externality. In the second period, the planner decides whether to make the AI algorithm available and how many licenses to offer.

As in the model without beta testing, we compute the allocations that maximize the social surplus $\int_0^1 \mathcal{U}_i di + \mathcal{V}$. With quasi-linear utility, this problem is equivalent to maximizing efficiency. Any distribution of utilities can be achieved using lump-sum transfers.

We begin by describing the solution to the second-period problem, contingent upon the choices made in the first period about ℓ and μ_1.

Social problem, second period Development costs are incurred in the first period. If $\mu_1 > 0$, then the planner learns the externality of the AI algorithm, $\phi(\ell)$. If $\mu_1 = 0$, then the planner faces the same uncertainty about the AI algorithm's potential externalities, $\mathbb{E}[\phi(\ell)^2] = \sigma^2(\ell)$ as in the model without beta testing.

The expected social welfare in the second period, considering the available information, is given by:

$$\mathcal{W}_2 = \begin{cases} Ny + \left[u\left(\ell\right) - \left(N+1\right)\phi\left(\ell\right)^2\right]\mu_2^2 & \text{if } \mu_1 > 0, \\ Ny + \left[u\left(\ell\right) - \left(N+1\right)\sigma^2\left(\ell\right)\right]\mu_2^2 & \text{if } \mu_1 = 0. \end{cases}$$

We now determine the optimal μ_2. If $\mu_1 > 0$, it is optimal to make the algorithm available to the entire population, $\mu_2 = N$, if $u(\ell) - (N+1)\phi(\ell)^2 \geq 0$ and to not release the AI algorithm otherwise ($\mu_2 = 0$). If $\mu_1 = 0$, then $\mu_2 = N$ if $u(\ell) - (N+1)\sigma^2(\ell) \geq 0$ and $\mu_2 = 0$ otherwise.

In period two, the planner only releases AI algorithms that are socially beneficial, taking into account the externality on the entire population, $(N+1)\phi(\ell)^2$. In contrast, the developer considers only its own loss of utility due to externality, $\phi(\ell)^2$. This difference implies that the developer is willing to commercialize AI algorithms that are detrimental to society.

Proposition 3 (Socially optimal restrictions on algorithm release)**.** *The central planner does not release AI algorithms that would be commercialized in an unregulated equilibrium under two circumstances:*

1. *If $\mu_1 > 0$, the externality on the population are larger than the social benefits of implementing the AI algorithm, but the private benefits to the developer of implementing the algorithm are positive:*

$$\frac{u(\ell)}{N+1} < \phi(\ell)^2 \leq u(\ell).$$

2. *If $\mu_1 = 0$, the expected negative externalities on the population are larger than the benefits of implementing the AI algorithm, while the private expected benefits to the developer of implementing the algorithm are positive:*

$$\frac{u(\ell)}{N+1} < \sigma^2(\ell) \leq u(\ell).$$

The resulting social welfare in period two is given by:

$$\mathcal{W}_2^*\left(\ell,\mu_1\right) = Ny + \max\left\{u(\ell) - (N+1)\left[\phi(\ell)^2\mathcal{I}(\mu_1 > 0) + \sigma^2(\ell)(1-\mathcal{I}(\mu_1 > 0))\right], 0\right\}N^2,$$

where the * indicates that the value function has been maximized with respect to the choice of price and implementation in period two. As before, we assume that $\phi(\ell)$ is such that there is a strictly positive probability that both $u(\ell) - (N+1)\phi(\ell)^2 > 0$ and $u(\ell) - (N+1)\phi(\ell)^2 < 0$. This assumption means that the probability that the AI algorithm is implemented in the second period, given the information obtained in the first period, is strictly positive but less than one.

Lemma 2 (Social benefits of beta testing in period one)**.** *It is not optimal to choose* $\mu_1 = 0$*. Expected social welfare is higher in the second period when there is beta testing in the first period:*

$$\mathbb{E}[\mathcal{W}_2^*(\ell, \mu_1)] > \mathcal{W}_2^*(\ell, 0), \textit{if} \quad \mu_1 > 0.$$

Proof. Let μ_2^0 denote the optimal choice of μ_2 when $\mu_1 = 0$. Note that μ_2^0 is not state-contingent, it's either $\mu_2^0 = N$ or $\mu_2^0 = 0$, depending on the uncertainty regarding externality. Then, if $\mu_1 > 0$:

$$\begin{aligned}
\mathbb{E}[\mathcal{W}_2^*(\ell, \mu_1)] &= Ny + \mathbb{E}\left\{\max\left[\left(u(\ell) - (N+1)\phi(\ell)^2\right) N^2, 0\right]\right\} \\
&> Ny + \mathbb{E}\left\{\left[u(\ell) - (N+1)\phi(\ell)^2\right] (\mu_2^0)^2\right\} \\
&= Ny + \left[u(\ell) - (N+1)\sigma^2(\ell)\right] (\mu_2^0)^2 \\
&= \mathcal{W}_2^*(\ell, 0),
\end{aligned}$$

where the strict inequality follows from the fact that the probability that the AI algorithm is implemented in period two, given the information obtained in period one, is strictly positive but less than one. □

Social problem, first period The overall expected social welfare is given by

$$\mathcal{W} = (1-\beta)\left[Ny + \left\{u(\ell) - (N+1)\sigma^2(\ell)\right\}\mu_1^2\right] + \beta\mathbb{E}[\mathcal{W}_2^*(\ell, \mu_1)] - f(\ell).$$

Setting $\mu_1 = 0$ is never optimal. It is always better to set μ_1 to an infinitesimal value to generate information that can be used in period two.

From the standpoint of the first period, it is optimal to set $\mu_1 = N$ if $u(\ell) - (N+1)\sigma^2(\ell) \geq 0$ and $\mu_1 = 0$ if $u(\ell) - (N+1)\sigma^2(\ell) < 0$. However, beta testing in the first period generates information that is valuable in the second period (see Lemma 2), i.e., if $\mu_1 > 0$

$$\mathbb{E}[\mathcal{W}_2^*(\ell, \mu_1)] > \mathcal{W}_2^*(\ell, 0).$$

Just like in the unregulated equilibrium, the discontinuity in information generation from $\mu_1 = 0$ to $\mu_1 > 0$, implies that the problem may have a supremum but not a maximum. As before, we consider an *experimentation solution*: the planner makes AI licenses available to an infinitesimal fraction of households to test the algorithm and then decides whether to release the algorithm given the information revealed in period two.

Proposition 4 (Socially optimal implementation of an AI algorithm in periods one and two)**.** *Given ℓ, the social optimal number of users in the first period (μ_1^*) involves:*

1. *Beta testing ($\mu_1^* \downarrow 0$) if uncertainty is sufficiently large*

$$\sigma^2(\ell) > \frac{u(\ell)}{N+1}.$$

2. *Immediate implementation ($\mu_1^* = N$) if uncertainty is sufficiently small*

$$\sigma^2(\ell) \leq \frac{u(\ell)}{N+1}.$$

In both cases, upon learning the externality consequences of the AI, the planner:

1. *Does not implement the AI algorithm ($\mu_2^* = 0$) if the externalities are sufficiently large*

$$\phi(\ell)^2 > \frac{u(\ell)}{N+1}.$$

2. *Implements the AI algorithm ($\mu_2^* = N$) if the externalities are sufficiently small*

$$\phi(\ell)^2 \leq \frac{u(\ell)}{N+1}.$$

The planner either implements beta testing or releases the algorithm to the population in the first period. However, the planner always adopts a more cautious stance than the developer when deciding whether to beta test rather than make the algorithm available to the whole population. There are AI novelty levels for which the developer prefers an immediate release to the general public, while the planner opts for beta testing.

Upon learning in period one the externality of the AI algorithm, there are algorithms that the developer would find privately beneficial to continue commercializing in the second period that the planner withdraws from the market. Both of these observations stem from the fact that the planner considers the externalities affecting the entire population, while the developer is only concerned with the impact of the externality on its own utility.

In summary, because the planner considers the impact of externalities on the entire population, it is more cautious than the developer in the sense that it implements beta testing for externalities more often than the developer. The planner is also more conservative in releasing the algorithm in the second period.

Proposition 5 (Caution in beta testing and implementation)**.** *Fix ℓ. In period one:*

1. *If uncertainty is substantial, $\sigma^2(\ell) \geq u(\ell)$, both the planner and the developer agree to beta test.*
2. *If uncertainty is moderate, $\frac{u(\ell)}{N+1} < \sigma^2(\ell) < u(\ell)$, the planner and the developer disagree. It is optimal for the planner to do beta testing but the developer finds full-scale implementation without testing privately optimal.*
3. *If uncertainty is low, $\sigma^2(\ell) \leq \frac{u(\ell)}{N+1}$, both the planner and the developer agree to release the algorithm to the entire population without beta testing.*

In period two:

1. *If externalities are substantial, $\phi(\ell)^2 \geq u(\ell)$, both the planner and the developer agree to withdraw the algorithm from the market.*
2. *If externalities are moderate, $\frac{u(\ell)}{N+1} < \phi(\ell)^2 < u(\ell)$, the planner wants to withdraw the algorithm from the market, but the developer does not.*

3. *If externalities are low, $\phi(\ell)^2 \leq \frac{u(\ell)}{N+1}$, both the planner and the developer agree to release the algorithm to the entire population.*

Surprisingly, in contrast to the model without beta testing, the first best can feature a higher novelty level, ℓ, than the unregulated equilibrium. In the model with beta testing, the planner can be cautious in two ways. The first is choosing a lower, less risky novelty level ℓ. The second is beta testing and withdrawing the algorithm when the net social benefits are negative. The planner withdraws the algorithm from the market more often than the developer. Because it exercises caution in testing and implementing, the planner might prefer a higher novelty level. We show an example of this possibility in the Appendix B.

Table 1: Testing or releasing in the first period, model with externalities

Uncertainty	**Low**	**Medium**	**High**
$\sigma^2(\ell)$	$\sigma^2(\ell) \leq \frac{u(\ell)}{N+1}$	$\frac{u(\ell)}{N+1} \leq \sigma^2(\ell) \leq u(\ell)$	$\sigma^2(\ell) \geq u(\ell)$
Developer	release	release	test
Planner	release	test	test

Table 1 compares the decision to test the algorithm in the first period or release it to the entire population. When the algorithm is similar to the status quo (ℓ is low), there is low uncertainty about its external impacts, and the developer and the planner concur that it is optimal to release it immediately. When ℓ significantly deviates from the status quo so that uncertainty about external effects is high, there is a unanimous decision that the algorithm should undergo testing to assess its suitability for release. There is disagreement in situations with moderate uncertainty levels: the developer releases the algorithm without prior testing, whereas it is socially optimal to test the algorithm to evaluate whether it should be released.

Table 2: Release decisions in the second period, model with externalities

Externality	**Low**	**Medium**	**High**
$\phi(\ell)^2$	$\phi(\ell)^2 \leq \frac{u(\ell)}{N+1}$	$\frac{u(\ell)}{N+1} \leq \phi(\ell)^2 \leq u(\ell)$	$\phi(\ell)^2 \geq u(\ell)$
Developer	release	release	not release
Planner	release	not release	not release

Table 2 compares the developer's and planner's decision to release the algorithm in the second period. Since the algorithm was either tested or released to the entire population in the first period, its external effects are known in the second period. The developer and planner align their release decisions when external effects are low or high. However, there is disagreement when external effects are in an intermediate range: the developer opts to release the algorithm, whereas the planner chooses not to. This disparity occurs because the developer disregards the external effects of the algorithm on the population.

Tables 1 and 2 show that simply banning the development of algorithms that pose a high risk of adverse external effects is insufficient to implement the social optimum.

4 Regulating AI

In this section, we use the model with beta testing to study the implications of two forms of regulation.[2] The first regulation is to control beta testing in period one

[2] Recall that the model without beta testing is a particular case of the general model with $\beta = 0$.

and make the release of the algorithm conditional on the test results. The second is making developers liable for the external effects of their algorithms.

4.1 Beta testing with conditional approval

Suppose the regulator mandates either beta testing or immediate release in period one and approves the implementation of the technology in period two only if

$$\phi^2 \leq \frac{u(\ell)}{N+1}.$$

For a given ℓ, conditional approval in period two generates lower ex-ante uncertainty about the effects of the externality

$$\varsigma^2(\ell) = \int_{-\sqrt{\frac{u(\ell)}{N+1}}}^{\sqrt{\frac{u(\ell)}{N+1}}} \phi^2 dG_\ell(\phi) \leq \sigma^2(\ell),$$

where G_ℓ denotes the CDF of $\phi(\ell)$. The ex-ante uncertainty about the externality at time two is given by the variance of the externality conditional on approval multiplied by the probability of approval in the second period.

The following proposition shows that this popular policy proposal does not implement the social optimum.

Proposition 6 (Regulatory approval of algorithm release)**.** *Suppose the regulator controls whether an algorithm is implemented in both periods. Suppose, furthermore, that ex-ante uncertainty $\varsigma^2(\ell)$ is increasing in ℓ. Then, the developer chooses a novelty level higher than the social optimum.*

To streamline the exposition, we relegate the proof to the appendix. The intuition for this proposition is that, for a given ℓ that is worthwhile for the developer to implement, the developer's utility is higher than social welfare, and this difference increases in ℓ. This difference in objectives occurs because the developer cares relatively less about the externality than the regulator.

4.2 Developers are liable for externalities

Suppose that the regulator allows the developer to freely choose the novelty level and whether to implement the AI algorithm but makes the developer liable in each period for any negative externalities. This policy means that the developer is forced to pay:

$$\tau_t(\phi(\ell), \mu_t) = N\phi(\ell)^2 \mu_t^2,$$

where μ_t denotes the number of households to whom the developer sells licenses.

In this case, the utility of the developer is given by

$$\mathcal{V} = (1-\beta)\left(\begin{cases} \mu_1 p_1 - \sigma^2(\ell)\mu_1^2 - \mathbb{E}[\tau_t(\phi(\ell), \mu_1)], & \text{if } p_1 \leq u(\ell)\mu_1 \\ 0, & \text{if } p_1 > u(\ell)\mu_1 \end{cases}\right) + \beta\mathbb{E}[\mathcal{V}_2] - f(\ell),$$

where

$$\mathcal{V}_2 = \begin{cases} \mu_2 p_2 - \phi(\ell)^2 \mu_2^2 - \tau_t(\phi(\ell), \mu_2), & \text{if } p_2 \leq u(\ell)\mu_2 \quad \text{and} \quad \mu_1 > 0, \\ \mu_2 p_2 - \sigma^2(\ell)\mu_2^2 - \mathbb{E}[\tau_t(\phi(\ell), \mu_2)], & \text{if } p_2 \leq u(\ell)\mu_2 \quad \text{and} \quad \mu_1 = 0. \\ 0, & \text{if } p_2 > u(\ell)\mu_2. \end{cases}$$

It is still optimal for the AI developer to set $p_t = u(\ell)\mu_t$. Replacing this price and the liability payments, we see that the utility of the developer coincides with the objective function of the social planner (up to a constant term) when choosing the novelty level and making implementation decisions:

$$\mathcal{V} = (1-\beta)[u(\ell) - (N+1)\sigma^2(\ell)]\mu_1^2 + \beta\mathbb{E}[\mathcal{V}_2] - f(\ell),$$

where

$$\mathcal{V}_2 = \begin{cases} [u(\ell) - (N+1)\phi(\ell)^2]\mu_2^2, & \text{if } \mu_1 > 0, \\ [u(\ell) - (N+1)\sigma^2(\ell)]\mu_2^2, & \text{if } \mu_1 = 0. \end{cases}$$

Private and social incentives become aligned when AI developers are liable for external effects. It follows that the privately optimal decisions coincide with the social optimum. We summarize these results in the following proposition.

Proposition 7 (Full liability)**.** *If the developer is liable for the algorithm's external effects, then private and social incentives are aligned. This alignment implies that the testing, implementation, and novelty level ℓ chosen by the developer are the same as in the first best.*

4.2.1 Limited Liability

The previous policy may require the developer to pay large sums. Suppose there is limited liability, in the sense that the liability payment each period cannot exceed the developer's revenue

$$\tau_t(\phi(\ell), \mu_t) \leq p_t \mu_t.$$

In this case, the taxes imposed by the regulator on the AI developer are:

$$\tau_t(\phi(\ell), \mu_t) = \min\{N\phi(\ell)^2 \mu_t^2, p_t \mu_t\}.$$

In this regulatory environment, it is still optimal for the developer to charge the maximum price $p_t = u(\ell)\mu_t$.

As before, releasing or beta testing the AI algorithm is always optimal. So, we only need to consider the case with full information in period two. This result implies that

$$\mathcal{V}_2^*(\ell, \mu_1) = [u(\ell) - \phi(\ell)^2]\mu_2^2 - \min\{N\phi(\ell)^2 \mu_2^2, u(\ell)\mu_2^2\}.$$

Suppose that $N\phi(\ell)^2 < u(\ell)$, then the developer decides to withdraw the AI algorithm from the market if $\phi(\ell)^2 > u(\ell)/(N+1)$ and to sell AI licenses to the whole population if $\phi(\ell)^2 \leq u(\ell)/(N+1)$.

Instead, if $N\phi(\ell)^2 \geq u(\ell)$, then the developer makes no profits from selling AI licenses but still suffers the external consequences of the AI algorithm. In this case, the developer withdraws the AI algorithm from the market. Importantly, note that if $N\phi(\ell)^2 \geq u(\ell)$, then $\phi(\ell)^2 > u(\ell)/(N+1)$. It follows that the social planner agrees to withdraw the AI algorithm from the market.

In sum, even in the presence of limited liability, making AI developers liable for the external costs of their algorithms is sufficient to align incentives *in the second period*. We summarize these results in the following proposition.

Proposition 8 (Limited liability)**.** *Suppose that the regulator makes AI developers liable for the externality of the AI algorithm subject to limited liability. Then, the developer and the regulator agree on the implementation strategy in period two, i.e.,*

1. *If externalities are substantial, $\phi(\ell)^2 > u(\ell)/(N+1)$, both the regulator and the AI developer agree to withdraw the algorithm from the market.*

2. *If externalities are low, $\phi(\ell)^2 \leq u(\ell)/(N+1)$, both the regulator and the AI developer agree to release the algorithm to the entire population.*

It follows from Proposition 8 that, with limited liability, $\mathcal{V}_2^*(\ell, \mu_1) = \mathcal{W}^*(\ell, \mu_1) - Ny$. However, note that because $\mathbb{E}[\tau_1(\phi(\ell), N)] < N\sigma^2(\ell)$, then incentives in the first period are not aligned.

In the presence of limited liability, the AI developer chooses a different level of ℓ than the planner. The AI developer sells licenses to the whole population if

$$u(\ell) > \sigma^2(\ell) + \mathbb{E}[\min\{N\phi(\ell)^2, u(\ell)\}]$$

and beta tests the AI algorithm if

$$u(\ell) \leq \sigma^2(\ell) + \mathbb{E}[\min\{N\phi(\ell)^2, u(\ell)\}].$$

The utility of the AI developer is given by

$$\mathcal{V} = (1-\beta)\max\{u(\ell) - \sigma^2(\ell) - \mathbb{E}[\min\{N\phi(\ell)^2, u(\ell)\}]\}N^2 + \beta[\mathcal{W}^*(\ell) - Ny] - f(\ell).$$

So, in general, making AI developers liable for external effects is insufficient to implement the social optimum in the presence of limited liability.

4.2.2 Limited liability with beta testing

Consider now an environment with limited liability in which the regulator always mandates beta testing for externalities and makes the developer liable for the external effects of the AI in period 2. This policy implements the first best in cases where immediate release is not socially optimal.

Proposition 9 (Limited liability and mandatory beta testing)**.** *Suppose that there is limited liability and the regulator always mandates beta testing. This regulatory environment does not implement the social optimum when beta testing is not optimal. When beta testing is socially optimal, private and social incentives are aligned: it is optimal for the AI developer to set the novelty level of their algorithms equal to the socially optimal novelty level. The AI developer beta tests the algorithm in the first period and releases the algorithm to the entire population in the second period if and only if externalities are low* $\phi^2 \leq u(\ell)/(N+1)$.

The cost of beta testing is missing the opportunity to release the algorithm to the entire population in the first period. As beta converges to one, this opportunity cost converges to zero. The social optimum can be implemented in this limit by making the developers responsible for any external effects with limited liability and requiring mandatory beta testing.

Given that the beta testing phase typically represents only a small fraction of the AI algorithm's usage period, enforcing mandatory beta testing is nearly optimal in real-world applications.

Table 3 summarizes the optimality properties of various forms of regulation, including those discussed in propositions 7, 8, and 9. The developer controls the choice of ℓ and the decision to release the algorithm in the second period in all scenarios. The first column indicates who controls the testing/release decision in the first period: either the developer freely decides whether to test or release (first row), or the regulator makes testing mandatory (second row). Imposing mandatory testing is a simple regulatory policy, but it creates some inefficiency when uncertainty is low. In

Table 3: Social optimality of different types of regulation, model with externalities

Testing	**Decision**	**No Liability**	**Limited Liability**	**Unlimited Liability**
Chosen by developer	ℓ	suboptimal	suboptimal	optimal
	$t = 1$ Testing/release	suboptimal	suboptimal	optimal
	$t = 2$ Release	suboptimal	optimal	optimal
Mandatory	ℓ	suboptimal	optimal†	optimal†
	$t = 1$ Testing/release	optimal†	optimal†	optimal†
	$t = 2$ Release	suboptimal	optimal	optimal

† when testing is socially optimal.

such scenarios, it's socially optimal to deploy the algorithm to the entire population in the first period instead of conducting beta testing.

Unlimited liability (fifth column) ensures social optimality when the developer makes all the decisions. The outcome is consistently suboptimal under no liability (third column). With limited liability, optimality is achieved when the testing is mandatory, provided that immediate release is not optimal.

A natural policy not included in Table 3 is for the regulator to impose the beta testing policy used in the social optimum. However, this policy does not achieve the first best in a setting with limited liability. The objectives of the developer and the social planner differ. As a result, when it is socially optimal to release the algorithm without testing in the first period, the developer chooses a value of ℓ that is not socially optimal.

4.3 Internalities versus externalities

We end this section by briefly comparing models with externalities and internalities, which are deviations from rationality that lead households to make choices that are not in their self-interest. Appendix A contains our analysis of the model with internalities.

In the model with externalities, the developer overlooks the external impacts on the broader population but still personally experiences these effects, just like any household. These external effects increase with the number of algorithm users. Consequently, when externalities are high, the developer is dissuaded from releasing the algorithm. This restraining factor is absent in the model with internalities. This absence results from the following natural assumption: the developer is not affected by internalities in its own choices, either because it does not use the AI algorithm or because it is more sophisticated than the households. Therefore, in the model with internalities and without regulation, the developer always sells AI licenses to the whole population in the first period and never withdraws the product in the second period.

5 Conclusion

In this paper, we study an environment with substantial uncertainty about the social costs and benefits of new AI technologies. We use this environment to assess different regulatory proposals put forth by policymakers in Europe and the United States.

We discuss optimal AI regulation in a setting with a single country. International coordination might be necessary to achieve a global optimum in a world with multiple nations.

Suppose externalities are local, that is, using an AI algorithm in one country does not impose externalities on other countries. National regulators can achieve a global optimum without limited liability by holding developers accountable for local externalities. With limited liability, local regulators must enforce optimal beta testing policies and make developers liable for externalities. These implementations do not require global coordination.

International cooperation is generally required when there is tax competition or global externalities, that is when using an AI algorithm in one country imposes externalities on other countries. This cooperation is more challenging when some governments pursue objectives that are different from social welfare (see Beraja, Kao, Yang, and Yuchtman, 2023 for empirical evidence along these lines).

Cooperation aside, the basic regulatory principle that emerges from our normative analysis is that, in an environment with limited liability, private and public incentives are approximately aligned when regulators mandate beta testing to identify externalities or internalities and hold developers liable for the adverse impacts caused by their algorithms.

Working out how to measure externalities and internalities to implement this basic regulatory principle is an arduous task that requires substantial investment in expertise and computational resources by regulatory bodies. But as Isaac Asimov writes in his Foundation trilogy, ”It has been my philosophy of life that difficulties vanish when faced boldly.”

References

ACEMOGLU, D. AND P. RESTREPO (2022): "Tasks, Automation, and the Rise in U.S. Wage Inequality," *Econometrica*, 90, 1973–2016.

AGRAWAL, A., J. GANS, AND A. GOLDFARB (2022): *Prediction Machines: The Simple Economics of Artificial Intelligence*, Harvard Business Press.

BENIFEI, B. AND I.-D. TUDORACHE (2023): "Proposal for a Regulation of the European Parliament and of the Council on Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act)," Tech. rep., Committee on the Internal Market and Consumer Protection, European Union.

BERAJA, M., A. KAO, D. Y. YANG, AND N. YUCHTMAN (2023): "Exporting the Surveillance State via Trade in AI," Working paper, Brookings Center on Regulation and Markets.

BIDEN, J. R. (2023): "Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence," Tech. rep., The White House.

BROWN, N. AND T. SANDHOLM (2019): "Superhuman AI for Multiplayer Poker," *Science*, 365, 885–890.

BURSTEIN, A., E. MORALES, AND J. VOGEL (2019): "Changes in Between-Group Inequality: Computers, Occupations, and International Trade," *American Economic Journal: Macroeconomics*, 11, 348–400.

CALLANDER, S. (2011): "Searching and Learning by Trial and Error," *American Economic Review*, 101, 2277–2308.

ELOUNDOU, T., S. MANNING, P. MISHKIN, AND D. ROCK (2023): "Gpts are Gpts: An Early Look at the Labor Market Impact Potential of Large Language Models," *arXiv preprint arXiv:2303.10130*.

EUROPEAN COMMISSION (2020): "Report on the Safety and Liability Implications of Artificial Intelligence, the Internet of Things and Robotics," Tech. rep., European Commission.

——— (2022): "Regulatory Framework Proposal on Artificial Intelligence," Tech. rep., European Commission.

FARBOODI, M. AND L. VELDKAMP (2021): "A Model of the Data Economy," Tech. rep., National Bureau of Economic Research.

FARHI, E. AND X. GABAIX (2020): "Optimal Taxation with Behavioral Agents," *American Economic Review*, 110, 298–336.

GITTINS, J. (1974): "A Dynamic Allocation Index for the Sequential Design of Experiments," *Progress in statistics*, 241–266.

HERRNSTEIN, R. J., G. F. LOEWENSTEIN, D. PRELEC, AND W. VAUGHAN JR (1993): "Utility Maximization and Melioration: Internalities in Individual Choice," *Journal of behavioral decision making*, 6, 149–185.

ILUT, C. AND R. VALCHEV (2023): "Economic Agents as Imperfect Problem Solvers," *The Quarterly Journal of Economics*, 138, 313–362.

JONES, C. I. (2023): "The AI Dilemma: Growth Versus Existential Risk," Tech. rep., Technical Report, Stanford GSB. Mimeo.

JONES, C. I. AND C. TONETTI (2020): "Nonrivalry and the Economics of Data," *American Economic Review*, 110, 2819–2858.

LANGLOTZ, C. P., B. ALLEN, B. J. ERICKSON, J. KALPATHY-CRAMER, K. BIGELOW, T. S. COOK, A. E. FLANDERS, M. P. LUNGREN, D. S. MENDELSON, J. D. RUDIE, ET AL. (2019): "A Roadmap for Foundational Research on Artificial Intelligence in Medical Imaging: From the 2018 NIH/RSNA/ACR/The Academy Workshop," *Radiology*, 291, 781–791.

MILGROM, P. AND C. SHANNON (1994): "Monotone Comparative Statics," *Econometrica: Journal of the Econometric Society*, 157–180.

SILVER, D., T. HUBERT, J. SCHRITTWIESER, I. ANTONOGLOU, M. LAI, A. GUEZ, M. LANCTOT, L. SIFRE, D. KUMARAN, T. GRAEPEL, ET AL. (2018): "A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go Through Self-play," *Science*, 362, 1140–1144.

SILVER, D., J. SCHRITTWIESER, K. SIMONYAN, I. ANTONOGLOU, A. HUANG, A. GUEZ, T. HUBERT, L. BAKER, M. LAI, A. BOLTON, ET AL. (2017): "Mastering the Game of Go Without Human Knowledge," *Nature*, 550, 354–359.

THOMPSON, W. R. (1933): "On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples," *Biometrika*, 25, 285–294.

A Proof of Proposition 1

Define

$$\mathcal{O}(\ell,d)=\begin{cases}\{u(\ell)-\sigma^2(\ell)\}N^2-f(\ell) & \text{if } d=1,\\ \{u(\ell)-(N+1)\sigma^2(\ell)\}N^2-f(\ell) & \text{if } d=0.\end{cases}$$

If $d=1$, then $\mathcal{O}(\ell,1)$ is the objective function of the AI developer, whereas if $d=0$, then $\mathcal{O}(\ell,0)$ is the objective function of the social planner.

Define the maximum admissible level of novelty considered by the developer and the social planner $\overline{\ell}(d)$ for $d=0$ and $d=1$ respectively. Note that $\mathcal{O}(\ell,1)>\mathcal{O}(\ell,0)$. Given our assumptions on u,σ, and f, this condition implies that

$$\overline{\ell}(1)>\overline{\ell}(0).$$

This result shows that the social planner implements lower levels of novelty than the developer. In particular, because the social planner has a higher weight on the externality, it does not allow the implementation of any novelty level $\ell\in(\overline{\ell}(0),\overline{\ell}(1)]$.

Finally, let

$$\ell^*(d)=\arg\max_{\ell}\mathcal{O}(\ell,d)$$

be the optimal novelty level for the developer if $d=1$ and the social planner if $d=0$. If $\ell^*(1)\notin[0,\overline{\ell}(0)]$, then it immediately follows that $\ell^*(0)<\ell^*(1)$ since $\ell^*(0)\in[0,\overline{\ell}(0)]$.

Then, suppose that $\ell^*(1)\in[0,\overline{\ell}(0)]$. We first show that the function $\mathcal{O}(\ell,d)$ satisfies strict single crossing in (ℓ,d). Then, using the monotone comparative statics results in Milgrom and Shannon (1994), we find that $\ell^*(1)>\ell^*(0)$.

Take $\ell'>\ell$, we show that

$$\mathcal{O}(\ell',0)\geq\mathcal{O}(\ell,0)\Rightarrow\mathcal{O}(\ell',1)>\mathcal{O}(\ell,1).$$

Note that

$$\begin{aligned}
&\mathcal{O}(\ell',0) \geq \mathcal{O}(\ell,0) \\
\Leftrightarrow &\{u(\ell') - (N+1)\sigma^2(\ell')\}N^2 - f(\ell') \geq \{u(\ell) - (N+1)\sigma^2(\ell)\}N^2 + -f(\ell) \\
\Leftrightarrow &\{u(\ell') - \sigma^2(\ell')\}N^2 - f(\ell') - N(\sigma^2(\ell') - \sigma^2(\ell))N^2 \geq \{u(\ell) - \sigma^2(\ell)\}N^2 - f(\ell) \\
\Leftrightarrow &\mathcal{O}(\ell',1) - N(\sigma^2(\ell') - \sigma^2(\ell))N^2 \geq \mathcal{O}(\ell',0).
\end{aligned}$$

Since $\sigma^2(\ell') > \sigma^2(\ell)$ and $\varsigma^2(\ell') > \varsigma^2(\ell)$, then the previous expression implies that

$$\mathcal{O}(\ell',1) > \mathcal{O}(\ell',0).$$

Since $\mathcal{O}(\ell,d)$ satisfies the single-crossing property, the results in Milgrom and Shannon (1994) imply that $\ell^*(d)$ is increasing in d. In other words,

$$\ell^*(1) \geq \ell^*(0),$$

i.e., the developer chooses a higher novelty level than the social planner. We prove this result by contradiction. Suppose that $\ell^*(1) < \ell^*(0)$. Since $\ell^*(0)$ is optimal at $d = 0$, it must be that

$$\mathcal{O}(\ell^*(0),0) \geq \mathcal{O}(\ell^*(1),0).$$

Since $\mathcal{O}$ satisfies the single-crossing property, then it follows that

$$\mathcal{O}(\ell^*(0),1) > \mathcal{O}(\ell^*(1),1),$$

which is contradicts the fact that $\ell^*(1)$ is optimal at $d = 1$.

B Example where social optimum has higher novelty than unregulated equilibrium

In this appendix, we provide an example in which, by being more cautious in beta testing and implementation, the social planner opts for a higher level of novelty than the AI developer.

The numerical example is as follows. Suppose that $u(\ell) = 2\sqrt{\ell}$ and that $f(\ell) = \chi\ell^2/2$ with $\chi = 10$. In addition, assume that $\beta = 0.7$ and that $\phi(\ell)$ is such that

$$\phi(\ell) = \begin{cases} \varphi\ell^2, & \text{with prob. } \frac{1-\alpha}{2} \\ 0, & \text{with prob. } \alpha \\ -\varphi\ell^2, & \text{with prob. } \frac{1-\alpha}{2}. \end{cases}$$

We set $\varphi = 1.0079$. In this case:

$$\sigma^2(\ell) = (1-\alpha)\,\overline{\phi}\psi^2\varphi^4.$$

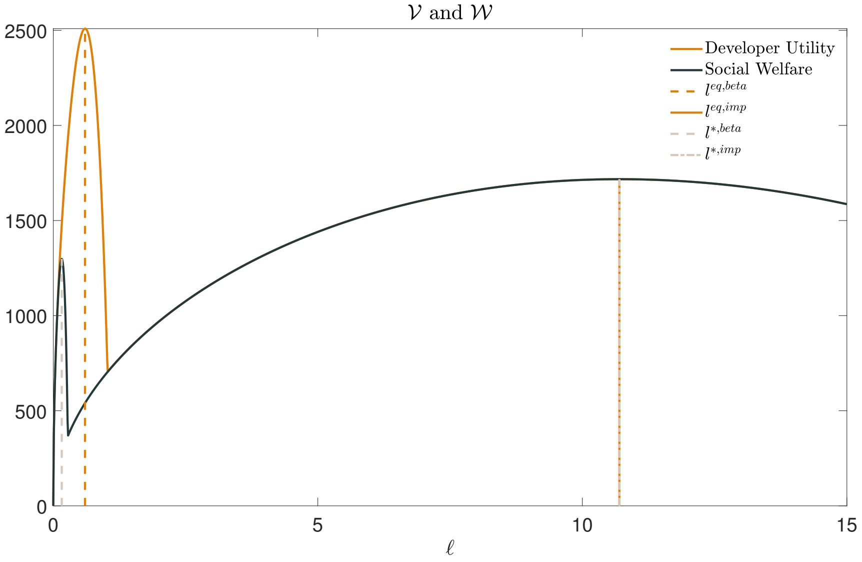

Figure 1: Example where social optimum has higher novelty than developer's optimum

In this case, the developer chooses a novelty level of 0.6 and immediately releases the algorithm to the whole population in period one. If the social planner was forced to release the algorithm to the whole population, it would choose a lower novelty level $\ell = 0.16$. However, by beta testing the algorithm in period one, the planner prefers a much higher novelty level: 10.7.

Suppose the developer was forced to beta test the algorithm in the first period; we see that they would choose the same novelty level as the planner. This result follows from the fact that, in this example, both the developer and planner agree to commercialize the AI at time 2 only if $\phi(\ell) = 0$. However, beta testing is privately suboptimal since it would cost the developer the profits earned in the first period.

C Proof of Proposition 6

Define

$$\mathcal{O}(\ell, d) = \begin{cases} (1-\beta)\{u(\ell) - \sigma^2(\ell)\}(\mu_1^*)^2 + \beta \int_{-\sqrt{\frac{u(\ell)}{N+1}}}^{\sqrt{\frac{u(\ell)}{N+1}}} [u(\ell) - hi^2] dG_\ell(\phi) - f(\ell) & \text{if } d = 1, \\ (1-\beta)\{u(\ell) - (N+1)\sigma^2(\ell)\}(\mu_1^*)^2 + \beta \int_{-\sqrt{\frac{u(\ell)}{N+1}}}^{\sqrt{\frac{u(\ell)}{N+1}}} [u(\ell) - (N+1)\phi^2] dG_\ell(\phi) - f(\ell) & \text{if } d = 0. \end{cases}$$

If $d = 1$, then $\mathcal{O}(\ell, 1)$ is the objective function of the AI developer, whereas if $d = 0$ then $\mathcal{O}(\ell, 0)$ is the objective function of the social planner. Let

$$\ell^*(d) = \arg\max_{\ell} \mathcal{O}(\ell, d),$$

be the optimal novelty level for the developer if $d = 1$ and the social planner if $d = 0$.

We first show that the function $\mathcal{O}(\ell, d)$ satisfies strict single crossing in (ℓ, d). Then, using the monotone comparative statics results in Milgrom and Shannon (1994), we find that $\ell^*(1) > \ell^*(0)$.

Take $\ell' > \ell$, we show that

$$\mathcal{O}(\ell', 0) \geq \mathcal{O}(\ell, 0) \Rightarrow \mathcal{O}(\ell', 1) > \mathcal{O}(\ell, 1).$$

Let $\alpha(\ell) = \mathbb{P}\left[\phi(\ell)^2 \leq \frac{u(\ell)}{N+1}\right]$ be the ex-ante probability that the AI algorithm is implemented in period 2. Note that

$$\begin{aligned}
&\mathcal{O}(\ell',0) \geq \mathcal{O}(\ell,0) \\
\Leftrightarrow &(1-\beta)\{u(\ell') - (N+1)\sigma^2(\ell')\}(\mu_1^*)^2 + \beta\{u(\ell')\alpha(\ell') - (N+1)\varsigma^2(\ell')\} - f(\ell') \\
&\geq (1-\beta)\{u(\ell) - (N+1)\sigma^2(\ell)\}(\mu_1^*)^2 + \beta\{u(\ell)\alpha(\ell) - (N+1)\varsigma^2(\ell)\} - f(\ell) \\
\Leftrightarrow &(1-\beta)\{u(\ell') - \sigma^2(\ell')\}(\mu_1^*)^2 + \beta\{u(\ell')\alpha(\ell') - \varsigma^2(\ell')\} - f(\ell') \\
&- (1-\beta)N(\sigma^2(\ell') - \sigma^2(\ell))(\mu_1^*)^2 - \beta N(\varsigma^2(\ell') - \varsigma^2(\ell)) \\
&\geq (1-\beta)\{u(\ell) - \sigma^2(\ell)\}(\mu_1^*)^2 + \beta\{u(\ell)\alpha(\ell) - \varsigma^2(\ell)\} - f(\ell) \\
\Leftrightarrow &\mathcal{O}(\ell',1) - (1-\beta)N(\sigma^2(\ell') - \sigma^2(\ell))(\mu_1^*)^2 - \beta N(\varsigma^2(\ell') - \varsigma^2(\ell)) \geq \mathcal{O}(\ell',0).
\end{aligned}$$

Since $\sigma^2(\ell') > \sigma^2(\ell)$ and $\varsigma^2(\ell') > \varsigma^2(\ell)$, it follows from the previous formula that

$$\mathcal{O}(\ell',1) > \mathcal{O}(\ell',0).$$

Because $\mathcal{O}(\ell,d)$ satisfies the single-crossing property, then the results in Milgrom and Shannon (1994) imply that $\ell^*(d)$ is increasing in d. In other words,

$$\ell^*(1) \geq \ell^*(0),$$

i.e. the developer opts for a higher novelty level than what the social planner. We prove this result by contradiction. Suppose that $\ell^*(1) < \ell^*(0)$. Since $\ell^*(0)$ is optimal at $d = 0$, it must be that

$$\mathcal{O}(\ell^*(0),0) \geq \mathcal{O}(\ell^*(1),0).$$

Since $\mathcal{O}$ satisfies the single-crossing property, then it follows that

$$\mathcal{O}(\ell^*(0),1) > \mathcal{O}(\ell^*(1),1),$$

which is contradicts the fact that $\ell^*(1)$ is optimal at $d = 1$.

Online Appendix
Regulation Artificial Intelligence

A A model with internalities

In this Appendix, we study a model where the adverse effects of AI algorithms take the form of internalities instead of externalities. We consider only a model with beta testing since the analogous model without beta testing is a particular case when $\beta = 0$.

A.1 Unregulated equilibrium

Household's problem The household utility is given by,

$$\begin{aligned}\mathcal{U}_i = (1-\beta)\left\{y + [u(\ell) - p_1] \times \mathcal{I}_{1,i} - \mathbb{E}[\phi(\ell)^2] \times \mathcal{I}_{1,i}\right\} \\ + \beta\mathbb{E}\left[y + [u(\ell) - p_2] \times \mathcal{I}_{2,i} - \phi(\ell)^2 \times \mathcal{I}_{2,i}\right],\end{aligned}$$

where $\mathcal{I}_{t,i}$ is an indicator function that takes the value one if the household uses the AI algorithm and zero otherwise. In this model, households potentially experience adverse effects, $\phi(\ell)$, whenever they use an AI algorithm with novelty ℓ. Households might be manipulated by the algorithm to act against their best interest because, in deciding to use the algorithm, they overlook its negative consequences. We formalize this idea by assuming that $\mathcal{U}_i$ is the household's "experienced utility," but that households base their choices on a different, misspecified, objective function that we refer to as the "decision utility":[3]

$$\mathcal{U}_i^s = (1-\beta)\left\{y + [u(\ell) - p_1] \times \mathcal{I}_{1,i}\right\} + \beta\mathbb{E}\left[y + [u(\ell) - p_2] \times \mathcal{I}_{2,i}\right].$$

[3] This terminology is common in the behavioral price theory literature, see, e.g., Farhi and Gabaix (2020).

To simplify the problem, we abstract from positive network externalities in AI usage. Introducing these network externalities would not alter our conclusions.

The household decides whether to purchase the AI algorithm to maximize $\mathcal{U}_i^s$. The resulting decision rule is to buy the algorithm whenever $p_t \leq u(\ell)$.

We maintain the assumptions about the $\phi(\ell)$ distribution previously discussed in Section 2.

The AI developer's problem In the first period, the AI developer chooses ℓ, how many AI licenses to offer for sale (μ_1), and the pricing of these licenses (p_1).

We proceed under the assumption that the developer does not use the algorithm. As a result, the developer is not personally impacted by the internalities generated by the AI algorithm. Expanding our analysis to consider scenarios where the algorithm's internalities also influence the developer is straightforward. However, such an extension would not significantly alter our findings.

The utility of the developer in the second period is:

$$\mathcal{V}_2 = \begin{cases} \mu_2 p_2 & \text{if } p \leq u\left(\ell\right), \\ 0 & \text{if } p > u\left(\ell\right). \end{cases}$$

As before, the price that maximizes the developer's utility is $p_2 = u(\ell)$. Unlike in the model with externalities, here it is always optimal for the developer to release the AI algorithm to the entire population, $\mu_2 = N$. The developer benefits from the resulting sales revenue and does not experience any adverse effects. This setting contrasts with the externality model, where the developer bore its share of the adverse external effects.

The developer's maximized utility in the second period is

$$\mathcal{V}_2^*(\ell) = u(\ell)N.$$

Given that the developer always chooses to make the AI algorithm available to the entire population, the developer's utility in the second period is independent of the information about internalities gathered in the first period.

The developer's problem in period one is to choose ℓ, μ_1 and p_1 to maximize

$$\mathcal{V} = (1-\beta)\left(\begin{cases} \mu_1 p_1, & \text{if } p_1 \leq u(\ell) \\ 0, & \text{if } p_1 > u(\ell) \end{cases}\right) + \beta\mathbb{E}[\mathcal{V}_2^*(\ell)] - f(\ell).$$

The optimal price for the developer in the first period is $p_1 = u(\ell)$. As in the second period, it is optimal for the developer to release the AI algorithm to the whole population in the first period, $\mu_1 = N$.

Proposition 10 (Beta testing and algorithm release)**.** *In an unregulated equilibrium, the developer always foregoes beta testing and releases the AI algorithm to the entire population in the first period ($\mu_1^e = N$). Regardless of the information generated in the first period about internalities, the developer makes the algorithm available to the entire population in the second period ($\mu_2^e = N$).*

Because the developer releases the algorithm to the whole population at a price $p_t = u(\ell)$ in both periods, the optimal value of ℓ is the one that maximizes $u(\ell)N - f(\ell.)$ The first-order for the problem of maximizing the developer's utility is

$$u'(\ell^e)N = f'(\ell^e).$$

A.2 The planner's problem

The central planner maximizes the social surplus, $\int_0^1 \mathcal{U}_i di + \mathcal{V}$. In the first period, the planner decides the novelty level of the AI algorithm and the number of households that will have access to it.

If the AI algorithm is released in the first period, the planner learns its internalities. In the second period, the planner determines whether to release the AI algorithm and decides the number of licenses to distribute.

We examine the solution to the second-period problem, contingent on the decisions about ℓ and μ_1 made in the first period.

Social problem, second period If the algorithm is released in period one ($\mu_1 > 0$), then the planner learns the internal effects of the AI algorithm.

The expected social welfare in the second period, considering the available information, is given by:

$$\mathcal{W}_2 = \begin{cases} Ny + \left[u\left(\ell\right) - \phi\left(\ell\right)^2\right] \mu_2 & \text{if } \mu_1 > 0, \\ Ny + \left[u\left(\ell\right) - \sigma^2\left(\ell\right)\right] \mu_2 & \text{if } \mu_1 = 0. \end{cases}$$

The optimal value of μ_2 is as follows. If $\mu_1 > 0$ and $u(\ell) \geq \phi(\ell)^2$ or if $\mu_1 = 0$ and $u\left(\ell\right) \geq \sigma^2\left(\ell\right)$, then it is optimal to release the algorithm to the whole population in the second period, $\mu_2 = N$. In all other scenarios, it is not optimal to release the algorithm ($\mu_2 = 0$).

The planner chooses to release only those AI algorithms that provide a net benefit to the household, taking into account the internalities. In contrast, the developer maximizes its profits and disregards the internalities that affect the households. As a result, the developer may choose to release AI algorithms that reduce social welfare.

Proposition 11 (Optimal restrictions on algorithm release)**.** *The central planner refrains from releasing AI algorithms that would be deployed in an unregulated equilibrium under two circumstances:*

1. *When $\mu_1 > 0$ and the internal effects outweigh the benefits of using the AI algorithm,*

$$u(\ell) < \phi(\ell)^2.$$

.

2. *When $\mu_1 = 0$ and the expected social benefits of implementing the AI algorithm are negative:*

$$u(\ell) < \sigma^2(\ell).$$

The resulting social welfare in the second period is:

$$\mathcal{W}_2^*\left(\ell, \mu_1\right) = Ny + \max\left\{u(\ell) - \left[\phi(\ell)^2\mathcal{I}(\mu_1 > 0) + \sigma^2(\ell)(1 - \mathcal{I}(\mu_1 > 0))\right], 0\right\} N,$$

where the * indicates that the value function has been maximized with respect to the choice of price and implementation in period two.

Lemma 3 (Social benefits of beta testing in period one)**.** *Expected social welfare is higher in the second period when there is beta testing in the first period:*

$$\mathbb{E}[\mathcal{W}_2^*(\ell,\mu_1)] > \mathcal{W}_2^*(\ell,0), \text{ if } \mu_1 > 0.$$

Proof. Let μ_2^0 denote the optimal choice of μ_2 when $\mu_1 = 0$. Note that μ_2^0 is not state-contingent, it's either $\mu_2^0 = N$ or $\mu_2^0 = 0$, depending on the uncertainty regarding internalities. Then, if $\mu_1 > 0$:

$$\begin{aligned}\mathbb{E}[\mathcal{W}_2^*(\ell,\mu_1)] &= Ny + \mathbb{E}\left[\max\left\{(u(\ell) - \phi(\ell))\, N, 0\right\}\right] \\ &> Ny + \mathbb{E}\left[\left(u(\ell) - \phi(\ell)^2\right)\mu_2^0\right] = Ny + \left[u(\ell) - \sigma^2(\ell)\right]\mu_2^0 \\ &= \mathcal{W}_2^*(\ell,0).\end{aligned}$$

□

Social problem, first period The overall expected social welfare is given by

$$\mathcal{W} = (1-\beta)\left[Ny + \left\{u(\ell) - \sigma^2(\ell)\right\}\mu_1\right] + \beta\mathbb{E}[\mathcal{W}_2^*(\ell,\mu_1)] - f(\ell).$$

Setting $\mu_1 = 0$ is never optimal. It is always better to set μ_1 to an infinitesimal value to gather information for use in the second period.

From the standpoint of period one, alone, it is optimal to set $\mu_1 = N$ if $u(\ell) - \sigma^2(\ell) \geq 0$ and $\mu_1 = 0$ if $u(\ell) - \sigma^2(\ell) < 0$. However, conducting beta testing in the first period generates valuable information that can be used in the second period (see Lemma 3), i.e., if $\mu_1 > 0$

$$\mathbb{E}[\mathcal{W}_2^*(\ell,\mu_1)] > \mathcal{W}_2^*(\ell,0).$$

As in the model with externalities, the discontinuity in information generation from $\mu_1 = 0$ to $\mu_1 > 0$, implies that the problem may have a supremum but not a maximum. As before, we consider an *experimentation solution*: the planner makes AI licenses available to an infinitesimal subset of households to test the algorithm and then make informed decisions about its release in the second period.

Proposition 12 (Optimal implementation of an AI algorithm in periods one and two)**.** *Given ℓ, the social optimum number of users in period one (μ_1^*) involves:*

1. *Beta testing ($\mu_1^* \downarrow 0$) if uncertainty is sufficiently large*

$$\sigma^2(\ell) > u(\ell).$$

2. *Immediate implementation ($\mu_1^* = N$) if uncertainty is sufficiently small*

$$\sigma^2(\ell) \leq u(\ell).$$

In both cases, upon learning the internalities of the AI algorithm, the planner:

1. *Does not implement the AI algorithm ($\mu_2^* = 0$) if the internalities are sufficiently large*

$$\phi(\ell)^2 > u(\ell).$$

2. *Implements the AI algorithm ($\mu_2^* = N$) if the internalities are sufficiently small*

$$\phi(\ell)^2 \leq u(\ell).$$

As in the model with externalities, the planner is more cautious than the developer. This caution is evident in the planner's more frequent implementation of beta testing to assess internalities. Additionally, the planner demonstrates a more conservative approach when deciding whether to release the algorithm in the second period.

Proposition 13 (Caution in testing and implementation)**.** *Fix ℓ. In period one:*

1. *If uncertainty is substantial, $\sigma^2(\ell) > u(\ell)$, the planner and the developer disagree. It is optimal for the planner to do beta testing but the developer finds full-scale release without testing privately optimal.*

2. *If uncertainty is low, $\sigma^2(\ell) \leq u(\ell)$, both the planner and the developer agree to release the algorithm to the entire population without beta testing.*

In period two:

1. *If internalities are large, $u(\ell) < \phi(\ell)^2$, the planner wants to withdraw the algorithm from the market, but the developer does not.*

2. *If internalities are low, $\phi(\ell)^2 \leq u(\ell)$, both the planner and the developer agree to release the algorithm to the entire population.*

Table 4: Testing or releasing in the first period, model with internalities

Uncertainty	**Low**	**High**
$\sigma^2(\ell)$	$\sigma^2(\ell) \leq u(\ell)$	$\sigma^2(\ell) > u(\ell)$
Developer	release	release
Planner	release	test

Table 4 compares the testing or release decisions made by the planner and the developer in the first period. When the algorithm is similar to the status quo (ℓ is low), there is low uncertainty about its internalities. Under these conditions, both the developer and the planner agree that it is optimal to immediately release the algorithm to the whole population. When ℓ significantly deviates from the status quo so that there is substantial uncertainty about internality effects, there is disagreement: the developer releases the algorithm without prior testing, whereas it is socially optimal to test the algorithm to evaluate whether it should be released to the entire population.

Table 5: Release decisions in the second period, model with internalities

Uncertainty	**Low**	**High**
$\phi(\ell)^2$	$\phi(\ell)^2 \leq u(\ell)$	$\phi(\ell)^2 > u(\ell)$
Developer	release	release
Planner	release	not release

Table 5 compares the release decisions made by the developer and the planner in the second period. Given that the algorithm was either beta-tested or made available to everyone in the first period, its internal effects are known by the second period. The developer and planner agree to release the algorithm when internal effects are low. However, their decisions diverge when these effects are substantial: the developer favors fully releasing the algorithm, while the planner opts against it.

In the model with externalities, the developer overlooks the external impacts on the broader population but still personally experiences these effects, just like any household. These external effects increase with the number of algorithm users. Consequently, when externalities are high, the developer is dissuaded from releasing the algorithm, a restraining factor absent in this model. In the version of the model with internalities, the developer is not a user of the algorithm, so it is not affected by the internality.

A.3 Regulating AI

This section studies the regulatory approaches considered in the model with externalities.

A.3.1 Beta testing with conditional approval

Consider a situation where the regulator decides to either beta test or release in the first period and only authorizes the algorithm's release in the second period if

$$\phi^2 \leq u(\ell).$$

For a specific ℓ, conditional approval in the second period reduces the initial uncertainty about the internalities' impact

$$\varsigma^2(\ell) = \int_{-\sqrt{u(\ell)}}^{\sqrt{u(\ell)}} \phi^2 dG_\ell(\phi) \leq \sigma^2(\ell),$$

where G_ℓ is the CDF of $\phi(\ell)$.

The following proposition states that this policy proposal does not achieve the social optimum.

Proposition 14 (Regulatory approval of algorithm release)**.** *Suppose that the regulator controls the release of the algorithm in both periods. Suppose, furthermore, that ex-ante uncertainty $\varsigma^2(\ell)$ is increasing in ℓ. In this scenario, the developer chooses a higher level of novelty than is socially optimal.*

The detailed proof is provided in the appendix. The intuition for this proposition is that for any ℓ, the developer's utility is higher than social welfare, with this discrepancy increasing with ℓ. This divergence in objectives stems from the developer's disregard for the internality.

A.3.2 Developers are liable for negative consequences

Suppose the regulator allows the developer to choose the novelty level and whether to release the AI algorithm but holds the developer liable for any adverse internalities. Under this policy, the developer is obligated to pay:

$$\tau_t(\phi(\ell), \mu_t) = \phi(\ell)^2 \mu_t,$$

where μ_t represents the number of households to whom the developer sells licenses.

In this case, the utility of the developer is given by

$$\mathcal{V} = (1-\beta)\left(\begin{cases} \mu_1 p_1 - \mathbb{E}[\tau_t(\phi(\ell), \mu_1)], & \text{if } p_1 \leq u(\ell) \\ 0, & \text{if } p_1 > u(\ell) \end{cases}\right) + \beta \mathbb{E}[\mathcal{V}_2] - f(\ell),$$

where

$$\mathcal{V}_2 = \begin{cases} \mu_2 p_2 - \tau_t(\phi(\ell), \mu_2), & \text{if } p_2 \leq u(\ell) \quad \text{and} \quad \mu_1 > 0, \\ \mu_2 p_2 - \mathbb{E}[\tau_t(\phi(\ell), \mu_2)], & \text{if } p_2 \leq u(\ell) \quad \text{and} \quad \mu_1 = 0. \\ 0, & \text{if } p_2 > u(\ell). \end{cases}$$

When the AI developer is fully accountable for internalities, the optimal pricing strategy remains $p_t = u(\ell)\mu_t$. Replacing this price and the liability payments, we see that the utility of the developer coincides (up to a constant) with the social planner's objective function, so both choose the same novelty level and release decisions:

$$\mathcal{V} = (1-\beta)[u(\ell) - \sigma^2(\ell)]\mu_1 + \beta \mathbb{E}[\mathcal{V}_2] - f(\ell),$$

where

$$\mathcal{V}_2 = \begin{cases} [u(\ell) - \phi(\ell)^2]\mu_2, & \text{if } \mu_1 > 0, \\ [u(\ell) - \sigma^2(\ell)]\mu_2, & \text{if } \mu_1 = 0. \end{cases}$$

Private and social incentives become aligned when AI developers are fully liable for internalities. As a result, the decisions that are optimal from a private standpoint are also socially optimal. These findings are summarized in the following proposition:

Proposition 15 (Optimality of regulated equilibrium with full liability)**.** *If the developer is held accountable for the internal effects of the algorithm, then the private and social* *incentives become aligned. This alignment means that the developer's decisions regarding testing, release, and the selection of the novelty level coincide with the first best.*

Limited Liability Suppose there is limited liability, in the sense that the liability payment cannot exceed the developer's revenue

$$\tau_t(\phi(\ell), \mu_t) \leq p_t \mu_t.$$

In this case, the taxes imposed by the regulator on the AI developer are:

$$\tau_t(\phi(\ell), \mu_t) = \min\{\phi(\ell)^2 \mu_t, p_t \mu_t\}.$$

In this regulatory environment, it is still optimal for the developer to charge a price $p_t = u(\ell)$.

As before, it is never optimal to set $\mu_1 = 0$, it is always optimal to either beta-test the AI algorithm or release it to the entire population. So, we only need to consider the case where there is full information in the second period. This result implies that

$$\mathcal{V}_2^*(\ell) = [u(\ell) - \min\{\phi(\ell)^2, u(\ell)\}]\mu_2.$$

Suppose that $\phi(\ell)^2 \leq u(\ell)$, then the developer and the planner agree to release AI licenses to the whole population. On the other hand, if $\phi(\ell)^2 > u(\ell)$, then the planner does not release the algorithm. Since the developer makes no profits from selling AI licenses, it is indifferent between releasing the algorithm or not.

In conclusion, even in the presence of limited liability, making AI developers liable for the internalities of their algorithms is sufficient to align incentives *in the second period*. We summarize these results in the following proposition.

Proposition 16 (No restrictions on algorithm release with limited liability)**.** *Suppose that the regulator makes AI developers liable for the internalities of the AI algorithm subject to limited liability. Then, the developer and the regulator agree on the release strategy in period two, i.e.,*

1. *If internalities are significant, $\phi(\ell)^2 > u(\ell)$, both the regulator and the AI developer agree to withdraw the algorithm from the market.*
2. *If internalities are low, $\phi(\ell)^2 \leq u(\ell)$, both the regulator and the AI developer agree to make the algorithm available to the entire population.*

It follows from Proposition 16 that, with limited liability, $\mathcal{V}_2^*(\ell, \mu_1) = \mathcal{W}^*(\ell, \mu_1) - Ny$. However, because $\mathbb{E}[\tau_1(\phi(\ell), N)] < \sigma^2(\ell)N$, incentives in the first period are not aligned.

In the presence of limited liability, the AI developer chooses a different level of ℓ than the planner. The AI developer sells licenses to the whole population if

$$u(\ell) > \mathbb{E}[\min\{\phi(\ell)^2, u(\ell)\}]$$

and beta tests the AI algorithm if

$$u(\ell) \leq \mathbb{E}[\min\{\phi(\ell)^2, u(\ell)\}].$$

The utility of the AI developer is given by

$$\mathcal{V} = (1-\beta)\max\{u(\ell) - \mathbb{E}[\min\{\phi(\ell)^2, u(\ell), 0\}]\}N + \beta[\mathcal{W}^*(\ell, \mu_1) - Ny] - f(\ell).$$

So, with limited liability, making AI developers liable for internalities is insufficient to implement the social optimum

Limited liability with beta testing Consider now an environment with limited liability in which the regulator always mandates beta testing for internalities, and the developer is liable for the internalities of the AI in the second period. This policy implements the first best in cases where releasing the algorithm in the first period is not socially optimal.

Proposition 17 (Limited liability and beta testing)**.** *Suppose that there is limited liability and the regulator always mandates beta testing. This regulatory setting does not implement*

Table 6: Optimality of different types of regulation, model with internalities

Testing choice	**Decision**	**No Liability**	**Limited Liability**	**Unlimited Liability**
Developer	ℓ	suboptimal	suboptimal	optimal
	$t = 1$ Testing	suboptimal	suboptimal	optimal
	$t = 2$ Release	suboptimal	optimal	optimal
Regulator	ℓ	$\ell^{\text{eq}} > \ell^*$	optimal[†]	optimal
	$t = 1$ Testing	suboptimal	optimal	optimal
	$t = 2$ Release	suboptimal	optimal	optimal

[†] when testing is socially optimal.

the social optimum when beta testing is not optimal. When beta testing is socially optimal, private and social incentives are aligned. The AI developer beta tests the algorithm in the first period and releases the algorithm to the entire population if and only if internalities are low $\phi^2 \leq u(\ell)$. Furthermore, the developer's novelty choice is equal to the social optimum.

Table 6 summarizes the effectiveness of various regulatory approaches. In all scenarios, the developer makes the decision regarding the novelty level (ℓ) and the release of the algorithm. The first column indicates who makes the testing decision. Testing is either decided by the developer (first row) or mandated by the regulator (second row).

Unlimited liability (fifth column) ensures optimality regardless of who makes the testing decision. The outcome is consistently suboptimal under no liability (third column), even when the regulator controls testing. With limited liability, optimality is achieved when the regulator mandates unconditional testing, provided that immediate release is not optimal.