NBER WORKING PAPER SERIES

REGULATING ARTIFICIAL INTELLIGENCE

Joao Guerreiro
Sergio Rebelo
Pedro Teles

Working Paper 31921
http://www.nber.org/papers/w31921

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
November 2023, Revised September 2024

We thank Rodrigo Adao, Mark Aguiar, Marco Bassetto, Luis Garicano, Alessandro Pavan, and Pascual Restrepo for their comments and Ramya Raghavan for excellent research assistance. The views expressed herein are those of the authors and do not necessarily reflect the views of the National Bureau of Economic Research.

Regulating Artificial Intelligence
Joao Guerreiro, Sergio Rebelo, and Pedro Teles
NBER Working Paper No. 31921
November 2023, Revised September 2024
JEL No. H21, O33

ABSTRACT

Recent AI advancements promise substantial benefits but also pose significant societal risks. We show that an unregulated equilibrium is unlikely to mitigate these risks in a socially optimal way. Our analysis evaluates different regulatory approaches to AI, taking into account the presence of uncertainty and disagreement about the likelihood of misalignments that can generate societal costs. We characterize the Pigouvian taxes that deliver the efficient allocation. Our analysis emphasizes the practical difficulties of implementing these taxes when developers are protected by limited liability or when they hold different beliefs from the rest of society. We study the optimal time-consistent combination of testing and regulatory approval. While this policy doesn't guarantee efficient use of resources, it enables society to harness AI's benefits while mitigating its risks.

Joao Guerreiro
Department of Economics
University of California Los Angeles
Los Angeles, CA
jguerreiro@econ.ucla.edu

Sergio Rebelo
Northwestern University
Kellogg School of Management
Department of Finance
Leverone Hall
Evanston, IL 60208-2001
and CEPR
and also NBER
s-rebelo@northwestern.edu

Pedro Teles
Banco de Portugal
R. Francisco Ribeiro 2
1150 Lisboa
Portugal
and Univ Catolica Portuguesa and CEPR
pteles@ucp.pt

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.

AI algorithms can outperform humans in tasks such as playing chess (Silver et al., 2017) and Go (Silver et al., 2018), recognizing images (Langlotz et al., 2019), and predicting protein structures (Jumper et al., 2021). They have also made great strides in understanding, translating, and generating content (Eloundou et al., 2023) and solving business forecast problems (Agrawal et al., 2022). These and other breakthroughs promise significant societal benefits but also carry the risk of generating considerable societal costs.

Some of these costs stem from the alignment problem highlighted by Norbert Wiener (Wiener, 1960). Misalignments occur when AI algorithms optimize narrow goals that overlook the broader spectrum of human objectives. For instance, social media algorithms may prioritize engagement at the expense of user well-being (Russell et al., 2015, Amodei et al., 2016).

There is significant disagreement about AI's potential societal impact. For example, neural network pioneers Geoffrey Hinton and Yann LeCun offer starkly different perspectives. Hinton has recently expressed serious concerns about AI's possible negative societal consequences (see, e.g., Heaven, 2023), while LeCun believes that benefits far outweigh the risks, suggesting that fears about societal costs are exaggerated (see, e.g., Hart, 2024).

AI's rapid development has outpaced regulatory efforts, creating an urgent need to make private benefits consistent with societal interests. Europe and the United States are developing frameworks to address these challenges (European Commission, 2020, European Commission, 2022, Benifei and Tudorache, 2023, Biden, 2023).

Proposed approaches include mandatory testing, making developers liable for adverse outcomes, and classifying AI technologies into risk tiers, with a ban on those posing unacceptable risks.

To assess these regulatory approaches, it is useful to divide the potential costs of AI into two categories. The first is negative externalities, such as fueling political polarization, facilitating fraud, disseminating false information, jeopardizing financial stability, and weakening democracies (see, e.g., Acemoglu, 2021, and Beraja, Kao, Yang, and Yuchtman, 2023). The second is "internalities," a term coined by Herrnstein et al. (1993) for situations where people act against their self-interest because of misinformation, self-control issues, cognitive biases, and time inconsistency problems.

Our analysis explores different public policies toward AI using a model designed to capture the key features of the algorithms currently being developed: uncertainty and disagreement about the likelihood of their societal costs. We evaluate different regulatory approaches starting with scenarios with homogeneous expectations about AI's external effects and then addressing more complex situations with heterogeneous expectations.

In our model, an AI developer selects the innovation level of the algorithm relative to the current state of the art, facing ex-ante uncertainty about potential adverse external and internal effects. This uncertainty increases with the gap between the algorithm's level of innovation and the status quo. Experimentation, which we call beta testing, can reduce uncertainty about potential negative effects by releasing the algorithm to a small group before wider deployment. While commonly used to evaluate effectiveness, the beta testing we emphasize here assesses an algorithm's social costs.

Pigouvian taxes are often used to equate private and societal interests, but their application to AI is challenging. We consider two types of Pigouvian taxes. Ex-ante taxes charge developers the expected welfare cost of external effects. Ex-post

taxes charge them for the realized welfare cost of external effects. Although effective with homogeneous expectations, ex-post taxes fall short under limited liability, as developers ignore social damages exceeding their liability cap.

Ex-ante taxes are effective under limited liability but impractical due to their complexity, especially when regulators and developers have different expectations. Designing these taxes requires eliciting the developer's expectations, a challenging task since the developer has the incentive to conceal their true beliefs and feign pessimism about external effects.

In sum, the Pigouvian taxes that equate private and social incentives are too complex to implement in practice. What should regulators do? We discuss the optimal time-consistent combination of beta testing and regulatory approval. Although this policy does not ensure optimal resource use–developers may choose innovation levels that are not socially optimal–it allows society to benefit from a given AI's potential while mitigating its downsides.

In Section 2, we review the related literature. We discuss our benchmark model in Section 3. In Section 4, we evaluate various regulatory approaches. We study scenarios in which AI algorithms create internalities in Section 5. We conclude by discussing the implications of our model for the regulatory proposal under consideration in the U.S. and the European Union.

2 Related literature

Our paper relates to four important strands of research. The first is a nascent economics literature on AI regulation, which we briefly discuss below.

Blattner et al. (2021) explore a delegation game where an agent designs a prediction algorithm under regulatory constraints set by a principal. The agent can opt for complex prediction functions, which the principal can regulate using only simplified descriptions. The authors find that restricting agents to using fully transparent, sim-

ple algorithms is inefficient when the misalignment between the principal and agent is limited, and complex algorithms offer significantly better performance.

Acemoglu and Lensman (2023) study the optimal adoption of an AI technology that might cause a disaster. Uncertainty about this disaster is resolved over time, regardless of whether the technology is adopted. For this reason, there is an incentive to delay adoption.

Gans (2024) argues that when there is learning-by-doing about the social costs of AI, it may be optimal to accelerate AI adoption, especially when this adoption can be reversed if necessary.

Callander and Li (2024) consider a model in which firms have superior information about the impact of technology. They show that increased competition weakens the regulator's ability to obtain reliable information, reducing the likelihood of approving beneficial innovations. This outcome occurs because the regulator's interests are more closely in line with those of incumbent firms. They both benefit from the status quo and only want innovation that improves upon it. In contrast, new entrants, who have nothing to gain from the status quo, push for innovation approval whenever it serves their private interests, irrespective of broader societal consequences.

Our work makes three contributions to this literature. First, we study how uncertainty about internal and external effects and disagreement about the likelihood of these effects impact the design of optimal AI policies considering both full and limited liability settings. Second, we explore the important role that beta testing can play in mitigating the downside risk of AI. Third, we analyze the effectiveness of various regulatory approaches that the U.S. and the European Union are considering.

A second line of research related to our work studies the economic impact of AI (e.g., Burstein, Morales, and Vogel, 2019, Acemoglu and Restrepo, 2022, Jones, 2023, Ide and Talamas, 2023, and Martinez, 2021) and the critical role of data in AI

algorithms (e.g., Jones and Tonetti, 2020 and Farboodi and Veldkamp, 2021). Our contribution relative to this literature is to characterize the optimal policy to deal with the externalities and internalities stemming from AI algorithms and discuss potential implementation strategies.

A third related strand of research studies the value of experimentation (e.g., Callander, 2011 and Ilut and Valchev, 2023). Relative to this work, we consider the possibility of conducting beta tests before the full product launch, including determining the ideal sample size for such tests.

A fourth related research area 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). Relative to this literature, we consider a setting where private and social incentives diverge and offer policy solutions that equate these incentives in settings with homogeneous and heterogeneous expectations.

3 Benchmark model

We consider a two-period model with a continuum of identical households and a single AI developer. We interpret the first period as the short run and the second as the long run.[1] In our model, AI usage is risky because, as we discuss below, it can create misalignments that can produce considerable social costs. To evaluate these costs, the developer can test the algorithm in the first period. Based on the outcome of this test, they can then decide whether to release the algorithm in the second period.

In this section, we consider the case where expectations about AI-related external and internal effects are homogenous. We discuss the household problem, the AI developer's problem, and the unregulated equilibrium. Then, we characterize the

[1] We omit time subscripts throughout the text when this omission does not reduce clarity.

social optimum and compare it with the unregulated equilibrium.

3.1 Unregulated equilibrium

Household problem The economy has a continuum of households indexed by $j \in [0, N]$, where N denotes the total number of households in the population. Each household lives for two periods.

Household j's momentary utility in period t, $v_{j,t}$, has a quasi-linear form:

$$v_{j,t} = y_t + [u(\ell) - \mathbb{E}_t(i_t^2) - p_t] \times \mathcal{I}_{j,t} - \mathbb{E}_t(e_t^2), \tag{1}$$

where y_t is the fixed exogenous income earned in period $t = 1, 2$. $\mathbb{E}_1(\cdot)$ denotes the unconditional expectation at the beginning of period one and $\mathbb{E}_2(\cdot)$ denotes the expectation conditional on the information obtained at the end of period one.

The utility from using an algorithm with innovation level ℓ is given by $u(\ell)$. This utility function is increasing ($u' > 0$), concave ($u'' < 0$), and satisfies the Inada condition ($\lim_{\ell \downarrow 0} u'(\ell) = \infty$). We normalize $u(0) = 0$, where $\ell = 0$ represents the status quo level of algorithm development.

The indicator function $\mathcal{I}_{j,t}$ takes the value one if household j buys the AI license and zero otherwise. The mass of AI users at time t is $\mu_t \equiv \int_0^N \mathcal{I}_{j,t} dj$.

Alignment problems In the introduction, we discuss misalignments that arise when AI systems optimize for narrow goals that fail to encompass the complex and multi-faceted aims of humans. From the standpoint of designing public policy, it is useful to classify misalignments into two types.

The first is internal misalignments that impact the AI user's utility directly. For instance, AI algorithms might manipulate households into making decisions that lower their welfare. Households that do not use the algorithm are not affected by this type of misalignment. An internal misalignment, i_t, decreases momentary utility by i_t^2. Although households that use the algorithm cannot avoid this misalign-

ment, they consider its impact when making their purchase decision. Therefore, in equation (4), the utility of buying the algorithm is reduced by the expected welfare cost of the internal misalignment, $\mathbb{E}(i_t^2)$. In Section 5, we explore a scenario where, due to behavioral biases, the household neglects the potential internal effects of the algorithm when deciding whether to use it.

The second is external misalignments imposed on a given household through the use of the AI algorithm by other households. For example, using a form of social media powered by AI algorithms might polarize public opinion and distort the outcome of elections. An external misalignment, e_t, reduces momentary utility by e_t^2. This reduction is increasing in the measure of users, μ_t.

Households have control over the welfare impact of internal misalignments because they can choose not to buy the algorithm. However, they have no control over external misalignments, as these are influenced by whether others use the algorithm.

Short- and long-run misalignments We assume that the short-run impact of internal and external misalignments on utility is equal to the long-run impact plus a mean-zero random variable, ξ_x for $x \in \{i, e\}$:

$$i_1^2 = \phi_i(\ell)^2 + \xi_i, \qquad i_2^2 = \phi_i(\ell)^2,$$
$$e_1^2 = \phi_e(\ell)^2\mu_1 + \xi_e, \qquad e_2^2 = \phi_e(\ell)^2\mu_2.$$

The random variables ξ_x capture the idea that the full consequences of AI usage may not be fully realized in the short run but emerge over the long run.

For each innovation level ℓ, $\phi_x(\ell)$, $x \in \{i, e\}$, are random variables. Positive and negative values of $\phi_x(\ell)$ represent undesirable misalignments between the user objectives and AI outcomes.

We assume that the distributions of $\phi_x(\ell)$, for $x \in \{i, e\}$, satisfy two properties. First, the expected value of $\phi_x(\ell)$ is zero:

$$\mathbb{E}_1[\phi_x(\ell)] = 0.$$

Second, the uncertainty about the potential misalignments increases with the innovation level, ℓ. Let $\sigma_x^2(\ell)$ denote the uncertainty about the potential misalignment of an algorithm with innovation level ℓ:

$$\sigma_x^2(\ell) = \mathbb{E}_1\left[\phi_x(\ell)^2\right].$$

We assume that $\sigma_x^2(\ell)$ is increasing and convex in ℓ and that there is no uncertainty about the status quo: $\sigma_x^2(0) = 0$.

Information from beta testing Upon testing or releasing the algorithm in period one, society receives signals regarding the algorithm's internal and external misalignments. For simplicity, instead of specifying the underlying distributions of the random variables ξ_x, we model the impact of this new information on the posterior beliefs. We assume that if $\mu_1 > 0$, then the posteriors for the expected value and the variance of ℓ are given by

$$\mathbb{E}_2[\phi_x(\ell)] = \hat{\phi}_x, \qquad \mathbb{VAR}_2(\phi_x(\ell)) = \hat{\sigma}_x^2(\ell) < \sigma_x^2(\ell)$$

Beta testing reduces uncertainty but does not eliminate it as long as $\hat{\sigma}_x^2(\ell) > 0$. Consequently, the decisions that affect long-run welfare must be made under some residual uncertainty.

To simplify, we consider the case in which the information generated by the beta test is independent of the number of people adopting the algorithm. In the Appendix, we consider the case in which testing might fail to produce any information about the internal and external effects, and the likelihood of this failure decreases as the number of participants increases.

The posterior beliefs satisfy the following conditions,

$$\mathbb{E}_1[\mathbb{E}_2(\phi_x)] = \mathbb{E}_1(\hat{\phi}_x) = 0 \quad \text{and} \quad \mathbb{E}_1[\mathbb{E}_2(\phi_x^2)] = \mathbb{E}_1[\hat{\phi}_x^2 + \hat{\sigma}_x^2(\ell)] = \sigma_x^2(\ell).$$

Household decisions Household j chooses whether to purchase an algorithm license in each period to maximize their expected lifetime utility, which is given by

$$\mathcal{U}_j = (1-\beta)v_{j,1} + \beta \mathbb{E}_1(v_{j,2}). \tag{2}$$

The household buys an AI license in period t if the expected private benefits, net of internal effects caused by the algorithm, exceed the price of the algorithm. In period one, this condition is

$$u(\ell) - \sigma_i^2(\ell) \geq p_1.$$

A similar condition applies in period two:

$$u(\ell) - \mathbb{E}_2[\phi_i(\ell)^2] \geq p_2.$$

The expected negative welfare consequences of internal misalignments reduce the price that the household is willing to pay for the algorithm in periods one and two.

The AI developer's problem There is a single AI developer who, at the beginning of period one, chooses the level of innovation of the algorithm, ℓ, and incurs a development cost, $f(\ell)$. This cost function is increasing and convex in ℓ, with $f(0) = 0$.

In period one, the developer can beta test the algorithm by releasing it to a subset of the population, $\mu_1 < N$. Alternatively, it can choose not to release the algorithm ($\mu_1 = 0$) or make it available to the whole population ($\mu_1 = N$). We assume that the decision to deploy the algorithm in period one can be reversed in period two. If this reversal occurs, the algorithm does not influence the utility in period two.

The developer's problem in period two At the beginning of period two, the developer decides whether to release the algorithm to the population, choosing the number of AI licenses to offer for sale (μ_2) and the price of each license (p_2). At the end of period two, uncertainty about internal and external misalignments is realized.

The developer's utility in the second period is,

$$\mathcal{V}_2 = \begin{cases} p_2\mu_2 - \mathbb{E}_2[\phi_e(\ell)^2]\mu_2, & \text{if } p_2 \leq u(\ell) - \mathbb{E}_2[\phi_i(\ell)^2], \\ 0, & \text{otherwise.} \end{cases}$$

We assume that the developer is immune to the algorithm's internal effect but experiences disutility from the externality in the same way households do.

If the developer markets the algorithm, the optimal license price is

$$p_2 = u(\ell) - \mathbb{E}_2[\phi_i(\ell)^2].$$

The developer, acting as a monopolist, sets the price to capture the expected consumer surplus.[2]

The cost of developing the algorithm is sunk because it was incurred at the beginning of period one. In period two, the developer releases the algorithm if the maximum price the household is willing to pay is greater than the reduction in the developer's utility caused by the externality associated with the algorithm, i.e., if $p_2 \geq \mathbb{E}_2[\phi_e(\ell)^2]$.

If the algorithm is released in period one ($\mu_1 > 0$), the posterior means of $\phi_x^2(\ell)$, for $x \in \{i, e\}$, are given by $\mathbb{E}_2[\phi_x^2(\ell)] = \hat{\phi}_x^2 + \hat{\sigma}_x^2(\ell)$. Using this information, the developer releases the algorithm in period two if

$$u(\ell) - [\hat{\phi}_i^2 + \hat{\sigma}_i^2(\ell)] \geq \hat{\phi}_e^2 + \hat{\sigma}_e^2(\ell),$$

otherwise the algorithm is not released ($\mu_2 = 0$).

If the algorithm was not released in period one ($\mu_1 = 0$), it is released in period two to the whole population ($\mu_2 = N$) if

$$u(\ell) - \sigma_i^2(\ell) \geq \sigma_e^2(\ell),$$

[2]This pricing strategy is a form of perfect price discrimination, which does not generate deadweight losses associated with the developer's market power, but simply redistributes resources from the households to the monopolist.

and not released otherwise ($\mu_2 = 0$).

Given that an algorithm with innovation level ℓ was developed and μ_1 licenses were sold in the first period, the optimized developer utility in period two is,

$$\mathcal{V}_2^*(\ell, \mu_1) = \begin{cases} \max\{u(\ell) - \sum_{x\in\{i,e\}}[\hat{\phi}_x^2 + \hat{\sigma}_x^2(\ell)], 0\}N, & \text{if } \mu_1 > 0, \\ \max\{u(\ell) - \sum_{x\in\{i,e\}}\sigma_x^2(\ell), 0\}N, & \text{if } \mu_1 = 0. \end{cases}$$

The asterisk indicates that the value function is evaluated in period two at the developer's optimal price and implementation strategy.

To make the problem interesting, we assume that the distribution of $\phi_x(\ell)$ is such that there is a strictly positive probability that both $u(\ell) > \sum_{x\in\{i,e\}}[\hat{\phi}_x^2 + \hat{\sigma}_x^2(\ell)]$, in which case the developer releases the algorithm, and $u(\ell) < \sum_{x\in\{i,e\}}[\hat{\phi}_x^2 + \hat{\sigma}_x^2(\ell)]$ in which case the algorithm is not released. This assumption means that the probability of the algorithm being implemented in period two is strictly positive but less than one.

The following lemma shows that, from the perspective of period two, there is a strictly positive value of having beta tested in period one. The intuition for this result is that the developer is better off because it can make decisions contingent on the acquired information.

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,*

$$\mathbb{E}_1[\mathcal{V}_2^*(\ell, \mu_1)] > \mathcal{V}_2^*(\ell, 0), \quad \textit{if} \quad \mu_1 > 0.$$

Proof. If $\mu_1 > 0$:

$$\begin{aligned}
\mathbb{E}_1[\mathcal{V}_2^*(\ell, \mu_1)] &= \mathbb{E}_1 \left(\max \left\{ u(\ell) - \sum_{x \in \{i,e\}} [\hat{\phi}_x^2 + \hat{\sigma}_x^2(\ell)], 0 \right\} N \right) \\
&> \max \left\{ u(\ell) - \mathbb{E}_1 \left(\sum_{x \in \{i,e\}} [\hat{\phi}_x^2 + \hat{\sigma}_x^2(\ell)] \right), 0 \right\} N \\
&= \max \left\{ u(\ell) - \sum_{x \in \{i,e\}} \sigma_x^2(\ell), 0 \right\} N = \mathcal{V}_2^*(\ell, 0).
\end{aligned}$$

The inequality holds because the expected value of the maxima is higher than the maximum of the expected value. The inequality is strict because the probability that the algorithm is implemented in period two, given the information obtained in period one, is strictly positive but less than one. □

The developer's problem in period one At the beginning of the first period, the developer chooses the level of innovation, ℓ. Then, they choose the number of licenses, μ_1, and the price per license, p_1. The developer's objective function is given by:

$$\mathcal{V} = (1 - \beta) \left(\begin{cases} p_1 \mu_1 - \sigma_e^2(\ell) \mu_1, & \text{if } p_1 \leq u(\ell) - \sigma_i^2(\ell) \\ 0, & \text{if } p_1 > u(\ell) - \sigma_i^2(\ell) \end{cases} \right) + \beta \mathbb{E}_1[\mathcal{V}_2^*(\ell, \mu_1)] - f(\ell).$$

The optimal price for the developer is the maximum price the household is willing to pay: $p = u(\ell) - \sigma_i^2(\ell)$.

From the perspective of period one, it is optimal to set $\mu_1 = N$ if $u(\ell) - \sigma_i^2(\ell) \geq \sigma_e^2(\ell)$ and $\mu_1 = 0$ if $u(\ell) - \sigma_i^2(\ell) < \sigma_e^2(\ell)$. 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 $u(\ell) <$

$\sum_x \sigma_x^2(\ell)$ 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.

Table 1 summarizes the developer's testing and release decisions. The developer may choose to withdraw the product from the market even if the expected value of the misalignment, $\hat{\phi}_x$, is relatively low in absolute value, as long as the residual uncertainty, $\hat{\sigma}_x^2$, remains significant. This result reflects the fact that the negative consequences are not fully known in the short run, leading households and developers to be cautious about the algorithm's long-run consequences.

Table 1 also summarizes the socially optimal testing and release decisions, which we now turn to.

3.2 The first-best solution (planner's problem)

We consider a central planner that can choose, in the first period, both the innovation level of the AI algorithm and the number of households that can use it. If the algorithm is implemented in the first period, the planner obtains information about its internal and external effects. In the second period, the planner decides whether to make the algorithm available and how many licenses to offer.

We define social welfare as the sum of the households' and developer's utilities, $\int_0^N \mathcal{U}_i di + \mathcal{V}$. With quasi-linear utility, maximizing this social-welfare function is equivalent to maximizing efficiency. Any distribution of utilities can be achieved using lump-sum transfers.

Table 1: Testing, release, and withdrawal decisions

Time 1

Uncertainty	**Low**	**Medium**	**High**
$\sigma_e^2(\ell)$	$\left[0, \frac{u(\ell)-\sigma_i^2(\ell)}{N+1}\right)$	$\left[\frac{u(\ell)-\sigma_i^2(\ell)}{N+1}, u(\ell)-\sigma_i^2(\ell)\right)$	$[u(\ell)-\sigma_i^2(\ell), \infty)$
Developer	release	release	test
Social optimum	release	test	test

Time 2

Expect. + res. uncert.	**Low**	**Medium**	**High**
ψ_e	$\left[0, \frac{u(\ell)-\psi_i}{N+1}\right)$	$\left[\frac{u(\ell)-\psi_i}{N+1}, u(\ell)-\psi_i\right)$	$[u(\ell)-\psi_i, \infty)$
Developer	release	release	withdraw
Social optimum	release	withdraw	withdraw

Notes: Here $\psi_x \equiv \hat{\phi}_x^2 + \hat{\sigma}_x^2(\ell)$ for $x \in \{i, e\}$ denotes the sum of expected damage and uncertainty.

To compute the socially optimal allocations, we start by describing the solution to the second-period problem, contingent upon the choices made in the first period about ℓ and μ_1.

The planner's problem in period two The expected social welfare in the second period, considering the available information, is given by:

$$\mathcal{W}_2 = \begin{cases} Ny_2 + \left[u\left(\ell\right) - [\hat{\phi}_i^2 + \hat{\sigma}_i^2(\ell)] - (N+1)\left[\hat{\phi}_e^2 + \hat{\sigma}_e^2(\ell)\right]\right]\mu_2 & \text{if } \mu_1 > 0, \\ Ny_2 + \left[u\left(\ell\right) - \sigma_i^2(\ell) - (N+1)\,\sigma_e^2\left(\ell\right)\right]\mu_2 & \text{if } \mu_1 = 0. \end{cases}$$

We now determine the optimal μ_2. If $\mu_1 > 0$, then the posteriors are given by $\mathbb{E}_2[\phi_x^2(\ell)] = \hat{\phi}_x^2 + \hat{\sigma}_x^2(\ell)$. In this case, releasing the algorithm is optimal if

$$\frac{u(\ell) - [\hat{\phi}_i^2 + \hat{\sigma}_i^2(\ell)]}{N+1} \geq \hat{\phi}_e^2 + \hat{\sigma}_e^2(\ell),$$

otherwise, $\mu_2 = 0$.

If $\mu_1 = 0$, then $\mu_2 = N$ if

$$\frac{u(\ell) - \sigma_i^2(\ell)}{N+1} \geq \sigma_e^2(\ell),$$

and otherwise $\mu_2 = 0$.

In period two, the planner only releases AI algorithms that are expected to be socially beneficial, taking into account the expected external effects on the entire population, $(N+1)\mathbb{E}_2[\phi_e(\ell)^2]$. In contrast, the developer considers only its own expected loss of utility due to the externality, $\mathbb{E}_2[\phi_e(\ell)^2]$. This difference implies that the developer is willing to commercialize AI algorithms that are detrimental to society.

The resulting social welfare in period two is given by:

$$\mathcal{W}_2^*\left(\ell, \mu_1\right) = Ny_2 + \begin{cases} \max\{u(\ell) - [\hat{\phi}_i^2 + \hat{\sigma}_i^2(\ell)] - (N+1)[\hat{\phi}_e^2 + \hat{\sigma}_e^2(\ell)], 0\}N, & \text{if } \mu_1 > 0 \\ \max\{u(\ell) - \sigma_i^2(\ell) - (N+1)\sigma_e^2(\ell), 0\}N, & \text{if } \mu_1 = 0, \end{cases}$$

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

We assume that there is a strictly positive probability that $u(\ell) - [\hat{\phi}_i^2 + \hat{\sigma}_i^2(\ell)] > (N+1)[\hat{\phi}_e^2 + \hat{\sigma}_e^2(\ell)]$, in which case it is optimal to release the algorithm, and $u(\ell) -$

$[\hat{\phi}_i^2 + \hat{\sigma}_i^2(\ell)] < (N+1)[\hat{\phi}_e^2 + \hat{\sigma}_e^2(\ell)]$, in which case it is not. This assumption means that the probability of releasing the algorithm in the second period, given the information obtained in the first period, is strictly positive but less than one.

The equivalent of Lemma 1 for the planner is as follows.

Lemma 2 (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:*

$$\mathbb{E}_1[\mathcal{W}_2^*(\ell, \mu_1)] > \mathcal{W}_2^*(\ell, 0), \textit{if} \quad \mu_1 > 0.$$

Proof. If $\mu_1 > 0$:

$$\begin{aligned}
\mathbb{E}_1[\mathcal{W}_2^*(\ell, \mu_1)] &= Ny_2 + \mathbb{E}_1\left(\max\left\{u(\ell) - [\hat{\phi}_i^2 + \hat{\sigma}_i^2(\ell)] - (N+1)[\hat{\phi}_i^2 + \hat{\sigma}_i^2(\ell)], 0\right\}N\right) \\
&> Ny_2 + \max\left\{u(\ell) - \mathbb{E}_1\left(\hat{\phi}_i^2 + \hat{\sigma}_i^2(\ell)\right) - (N+1)\mathbb{E}_1\left(\hat{\phi}_i^2 + \hat{\sigma}_i^2(\ell)\right), 0\right\}N \\
&= Ny_2 + \max\left\{u(\ell) - \sigma_i^2(\ell) - (N+1)\sigma_e^2(\ell), 0\right\}N = \mathcal{V}_2^*(\ell, 0).
\end{aligned}$$

□

The planner's problem in period one Expected social welfare is given by

$$\mathcal{W} = (1-\beta)\left[Ny_1 + \left\{u(\ell) - \sigma_i^2(\ell) - (N+1)\sigma_e^2(\ell)\right\}\mu_1\right] + \beta\mathbb{E}_1[\mathcal{W}_2^*(\ell, \mu_1)] - f(\ell).$$

We now consider the optimal choice of μ_1 for a given ℓ. 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

$$\frac{u(\ell) - \sigma_i^2(\ell)}{N+1} \geq \sigma_e^2(\ell)$$

and μ_1 equal to an infinitesimal value (the experimentation solution) otherwise.

The planner is more cautious than the developer when deciding between beta testing and releasing the algorithm to the general population. At certain innovation levels, the developer prefers an immediate release to the general public, while the planner opts for beta testing.

Upon obtaining information about the external misalignments of the AI algorithm in period one, there are algorithms that the planner withdraws from the market in the second period, but the developer finds privately beneficial to continue commercializing.

In summary, for a given ℓ, 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 in the first period more often than the developer. The planner is also more conservative in releasing the algorithm in the second period.

Table 1 compares the planner's and developer's 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 to the whole population in period one. 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 in period one 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 1 also 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, information about its misalignments is available in the second period. The developer and planner make the same release decisions when external effects are low or high. However, there is disagreement when exter-

nal 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.

Surprisingly, the social optimum can feature a higher innovation level, ℓ, than the unregulated equilibrium. With beta testing, the planner can be cautious in two ways. The first is choosing a lower, less risky innovation level ℓ. The second is beta testing and withdrawing the algorithm when the net social benefits are negative. Because it exercises more caution than the planner in testing and implementing for any given ℓ, the planner might prefer a higher innovation level. We show an example of this possibility in Appendix A.

4 Regulating AI

The level of social welfare in the unregulated equilibrium is lower than that in the social optimum because developers ignore AI's external impact on household welfare.

We now show that using Pigouvian taxes to align developers' incentives with societal interests is challenging because AI's external effects are uncertain, and there is disagreement about their probability distribution.

As discussed in the introduction, we consider two types of Pigouvian taxes: ex-post, which hold developers fully liable for the realized external damages, and ex-ante, which charge developers the expected external effects. While ex-post taxes are effective when expectations about external misalignments are homogeneous, they fail under limited liability scenarios because developers ignore social damages beyond their liability cap. Ex-ante taxes, though theoretically effective under limited liability, are impractical because it is difficult to take into account differing expectations between regulators and developers and it is challenging to enforce these taxes.

Given the impracticality of Pigouvian taxes for regulating AI, we discuss an al-

ternative approach that combines controlling beta testing and regulating approval. This second-best policy does not guarantee the efficient use of resources, as developers might still choose suboptimal levels of innovation. Still, it offers a pragmatic way to balance AI's potential benefits and risks.

4.1 Pigouvian taxes with homogeneous beliefs

We first consider ex-post and ex-ante Pigouvian taxes in an economy with homogeneous beliefs.

4.1.1 Ex-post Pigouvian taxes with full liability

We can align private and social incentives by imposing, at the end of each period, ex-post taxes on the developer (τ_t) that are equal to the welfare cost of the realized external effects imposed on the households:

$$\tau_1 = N \times e_1^2 = N \times [\phi_e(\ell)^2 \mu_1 + \xi_{e,1}],$$

$$\tau_2 = N \times e_2^2 = N \times \phi_e(\ell)^2 \mu_2.$$

The expected value of these ex-post taxes are

$$\mathbb{E}_1(\tau_1) = N \sigma_e^2(\ell) \mu_1,$$

$$\mathbb{E}_2(\tau_2) = N(\hat{\phi}_e^2 + \hat{\sigma}_e^2(\ell)) \mu_2.$$

In the presence of these taxes, the expected utility of the developer at the beginning of period one is

$$\mathcal{V} = (1-\beta) \left(\begin{cases} p_1 \mu_1 - \sigma_e^2(\ell) \mu_1 - \mathbb{E}_1(\tau_1), & \text{if } p_1 \leq u(\ell) - \sigma_i^2(\ell) \\ 0, & \text{if } p_1 > u(\ell) - \sigma_i^2(\ell) \end{cases} \right) + \beta \mathbb{E}_1(\mathcal{V}_2) - f(\ell),$$

where $\mathcal{V}_2$, the expected utility of the developer at the beginning of period two, is given by

$$\mathcal{V}_2 = \begin{cases} p_2\mu_2 - \mathbb{E}_2[\phi_e(\ell)^2\mu_2 - \tau_2], & \text{if } p_2 \leq u(\ell) - \mathbb{E}_2[\phi_i(\ell)^2], \\ 0, & \text{if } p_2 > u(\ell) - \mathbb{E}_2[\phi_i(\ell)^2]. \end{cases}$$

It is still optimal for the AI developer to charge the maximum price the household is willing to pay $p_t = u(\ell) - \mathbb{E}_t[\phi_i^2(\ell)]$. Replacing this price and the expected taxes into the utility of the developer, we see that it coincides with the objective function of the social planner up to a constant term:

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

where

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

With these taxes, $\mathcal{V}$ coincides with $\mathcal{W}$ up to the constant term $(1-\beta)Ny_1 + \beta Ny_2$. It follows that private and social incentives are in line, so privately optimal decisions coincide with the social optimum. We summarize these results in the following proposition.

Proposition 1 (Full liability)**.** *Private and social incentives coincide if the regulator levies an ex-post tax on the developer equal to the welfare cost of the algorithm's realized external effects. This concurrence implies that the developer's testing, implementation, and innovation decisions coincide with the socially optimal ones.*

One important element of this policy is that the developer is fully liable for the external effects caused by the algorithm. In practice, it is impossible to enforce full liability, so we now consider the more realistic case of limited liability.

4.1.2 Ex-post Pigouvian taxes with limited liability

To study the consequences of limited liability, we consider the simple case in which the taxes paid by the developer in each period cannot exceed their revenue:

$$\tau_t \leq p_t \mu_t.$$

In this scenario, the taxes levied by the regulator on the AI developer are:

$$\tau_1 = \min\{N(\phi_e(\ell)^2 \mu_1 + \xi_{e,1}), p_1 \mu_1\},$$

$$\tau_2 = \min\{N\phi_e(\ell)^2 \mu_2, p_2 \mu_2\}.$$

It is still optimal for the developer to charge $p_t = u(\ell) - \mathbb{E}_t[\phi_i(\ell)^2]$.

Under limited liability, the developer's optimal algorithm release policy differs from the social optimum. This divergence arises because the developer's potential losses are capped, encouraging it to release moderately risky algorithms relying on limited liability to protect itself if significant negative external effects occur.

Consider the problem in period two. The developer is willing to release the algorithm as long as

$$u(\ell) - \mathbb{E}_2[\phi_i(\ell)^2] > \mathbb{E}_2[\phi_e(\ell)^2] + N\mathbb{E}_2[\min\{\phi_e(\ell)^2, p_2\}],$$

while it is socially optimal to release the algorithm only if

$$u(\ell) - \mathbb{E}_2[\phi_i(\ell)^2] > (N+1)\mathbb{E}_2[\phi_e(\ell)^2].$$

As long as the probability that $\phi_e(\ell)^2 > p_2$ is strictly positive, the liability limit binds and $\mathbb{E}_2(\tau_2) < \mathbb{E}_2[N\phi_e(\ell)^2]$. In this case, the expected value of the taxes on the developer is lower than the expected social welfare cost of the externality.

The same logic implies that the developer may forgo beta testing in period one and release the algorithm immediately, knowing it is protected by limited liability if dire external effects materialize. As a result, the developer may act with less caution than would be socially optimal.

4.1.3 Ex-ante Pigouvian taxes

We now consider a scenario in which the regulator imposes the following ex-ante Pigouvian taxes at the beginning of each period

$$\tau_t^{ex-ante} = \mathbb{E}_t\left(N \times e_t^2\right) = N\mathbb{E}_t\left[\phi_e\left(\ell\right)^2\right]\mu_t$$

These taxes are euqla to the expected external welfare damage of the algorithm for households. If the developer decides not to release the algorithm in period t ($\mu_t = 0$), ex-ante taxes are zero.

Since $\tau_t^{ex-ante} = \mathbb{E}_t[\tau_t]$, then these ex-ante taxes affect decisions in the same way as the ex-post taxes when there is full liability. So, these taxes equate private and social incentives, resulting in a regulated equilibrium that coincides with the social optimum.

A key advantage of ex-ante taxes is that they implement the social optimum, even with limited liability. We have described the ex-ante taxes with full liability. Imposing limited liability means that

$$\tau_t^{ex-ante} = \min\left\{N\mathbb{E}_t\left[\phi_e\left(\ell\right)^2\right]\mu_t, p_t\mu_t\right\}.$$

Trivially, if limited liability is not binding, then private and social incentives coincide. But, what if limited liability binds? Since $p_t = u(\ell) - \mathbb{E}_t[\phi_i(\ell)^2]$, then limited liability binds whenever $u(\ell) - \mathbb{E}_t[\phi_i(\ell)^2] < N\mathbb{E}_t\left[\phi_e\left(\ell\right)^2\right]$. In that case, the developer makes zero profits from selling the algorithm. However, they still suffer from the externality created. It follows that whenever limited liability binds, developers strictly prefer not to release the algorithm. As it turns out, limited liability only binds whenever the regulator would strictly prefer not to release the algorithm too since $N\mathbb{E}_t[\phi_e(\ell)^2] < (N+1)\mathbb{E}_t[\phi_e(\ell)^2]$. So, whenever limited liability binds, both the developer and the regulator agree not to release the algorithm.

Implementing these taxes can lead to situations where developers must pay significant upfront taxes based on anticipated external misalignments that ultimately

do not occur. These situations can complicate the enforcement of ex-ante taxes. These challenges become even greater when the regulator and the developer have heterogeneous expectations with respect to external effects. We now turn to this case.

4.2 Pigouvian taxes with heterogeneous beliefs

Up to this point, we've assumed that beliefs about external effects are homogeneous. However, as discussed in the introduction, there are notable differences in beliefs regarding alignment problems.

We analyze how the two policies that implement the social optimum with homogeneous expectations, ex-post Pigouvian taxes with full liability and ex-ante Pigouvian taxes (with or without full liability), fare when beliefs are heterogeneous. For simplicity, we consider a scenario where households and the regulator share the same beliefs, which differ from the developer's.

We let

$$\mathbb{E}_1^d[\phi_x(\ell)^2] = \sigma_{d,x}^2(\ell), \quad \mathbb{E}_2^d[\phi_x(\ell)^2] = \hat{\phi}_{d,x}^2 + \hat{\sigma}_{d,x}^2(\ell)$$

denote the developer's beliefs and

$$\mathbb{E}_1^s[\phi_x(\ell)^2] = \sigma_{s,x}^2(\ell), \quad \mathbb{E}_2^s[\phi_x(\ell)^2] = \hat{\phi}_{s,x}^2 + \hat{\sigma}_{s,x}^2(\ell)$$

denote the beliefs held by the households and regulators.

4.2.1 Ex-ante Pigouvian taxes with heterogenous beliefs

Consider first the case in which ex-ante taxes are based on the regulator's beliefs.

$$\tau_1^{ex-ante} = N\sigma_{s,e}^2(\ell)\mu_1,$$

$$\tau_2^{ex-ante} = N(\hat{\phi}_{s,e}^2 + \hat{\sigma}_{s,e}^2(\ell))\mu_2.$$

The developer's expected utility at the beginning of period one is given by:

$$\mathcal{V} = (1-\beta)\left(\begin{cases} p_1\mu_1 - \sigma_{d,e}^2(\ell)\mu_1 - \tau_1, & \text{if } p_1 \leq u(\ell) - \sigma_{s,i}^2(\ell) \\ 0, & \text{if } p_1 > u(\ell) - \sigma_{s,i}^2(\ell) \end{cases}\right) + \beta\mathbb{E}_1^d(\mathcal{V}_2) - f(\ell),$$

where $\mathcal{V}_2$ is

$$\mathcal{V}_2 = \begin{cases} p_2\mu_2 - \mathbb{E}_2^d[\phi_e(\ell)^2\mu_2] - \tau_2, & \text{if } p_2 \leq u(\ell) - \mathbb{E}_2^s[\phi_i(\ell)^2], \\ 0, & \text{if } p_2 > u(\ell) - \mathbb{E}_2^s[\phi_i(\ell)^2]. \end{cases}$$

Because of their differing expectations, the developer's utility differs from the social welfare function, so these taxes do not fully equate private and social incentives. When developers are more optimistic about external effects than society, i.e., $\mathbb{E}_t^d[\phi_e(\ell)^2] < \mathbb{E}_t^s[\phi_e(\ell)^2]$, they have an incentive to release algorithms that society perceives as too risky.

Alternatively, we can design taxes that correct the difference in beliefs and implement the social optimum. These taxes are given by:

$$\tau_t^{ex-ante} = N\mathbb{E}_t^s\left[\phi_e\left(\ell\right)^2\right]\mu_t + \left(\mathbb{E}_t^s\left[\phi_e\left(\ell\right)^2\right] - \mathbb{E}_t^d[\phi_e(\ell)^2]\right)\mu_t$$

The first term internalizes the externality according to the regulator's expectations. The second term corrects for the difference in expectations between the developer and the regulator. The developer should face a higher tax when they are relatively optimistic, $\mathbb{E}_t^d[\phi_e(\ell)^2] < \mathbb{E}_t^s[\phi_e(\ell)^2]$, and a lower tax when they are relatively pessimistic, $\mathbb{E}_t^d[\phi_e(\ell)^2] > \mathbb{E}_t^s[\phi_e(\ell)^2]$.

These taxes equate private and social incentives to release the algorithm in both periods. However, they will not align development incentives because

$$\mathbb{E}_1^s\left[\max\left\{u\left(\ell\right)N - \left(N+1\right)\mathbb{E}_2^s\left[\phi_e\left(\ell\right)^2\right]N, 0\right\}\right] \neq$$
$$\mathbb{E}_1^d\left[\max\left\{u\left(\ell\right)N - \left(N+1\right)\mathbb{E}_2^s\left[\phi_e\left(\ell\right)^2\right]N, 0\right\}\right],$$

that is, the regulator and the planner have different beliefs in period one about what they will believe in period two.

Another significant problem with this policy is that the developer's expectations are generally unobservable. Suppose the regulator has to elicit these expectations. In that case, the developer has the incentive to misrepresent their expectations by

claiming pessimistic views regarding external effects, i.e., high values of $\mathbb{E}_t^d[\phi_e(\ell)^2]$ to receive a subsidy instead of paying a tax. Even if the regulator could accurately measure the developer's expectations, this policy would be difficult to design and enforce.

4.2.2 Ex-post Pigouvian taxes with full liability and heterogenous beliefs

When the developer is more optimistic (pessimistic) than society, the ex-post Pigouvian taxes that implement the social optimum are higher (lower) than the actual damages.

$$\tau_1 = N \times [\phi_e(\ell)^2 \mu_1 + \xi_{e,1}] + \left(\mathbb{E}_1^s \left[\phi_e \left(\ell \right)^2 \right] - \mathbb{E}_1^d [\phi_e(\ell)^2] \right) \mu_1$$

$$\tau_2 = N \times \phi_e(\ell)^2 \mu_2 + \left(\mathbb{E}_2^s \left[\phi_e \left(\ell \right)^2 \right] - \mathbb{E}_2^d [\phi_e(\ell)^2] \right) \mu_2$$

Like ex-ante Pigouvian taxes, these taxes would be difficult to design and even more challenging to enforce. Developers would still have an incentive to feign greater pessimism about external effects than the planner to minimize their tax burden. Additionally, while these taxes may align private and social incentives to release the algorithm, they do not align the choice of the level of innovation.

In summary, the conventional approach to dealing with externalities, Pigouvian taxes, is unworkable in an AI context.

4.3 Testing and approval policies

Considering the challenges associated with Pigouvian taxes, we now explore policies where the regulator can mandate beta testing in the first period and control whether the algorithm is released.

4.3.1 An optimal but time-inconsistent policy

There is a straightforward testing and approval policy that achieves the social optimum. The regulator announces that it will not approve any algorithm with an

innovation level ℓ different from the social optimum, ℓ^*,

$$\mu_1 = \mu_2 = 0 \text{ for all } \ell \neq \ell^*.$$

When a developer creates an algorithm with the socially optimal level of ℓ, the regulator applies the optimal beta testing policy (as outlined in Table 1) and, when there is testing, conditions the algorithm's release in period two on the test results.

Unfortunately, this policy is time-inconsistent because it requires the regulator to prohibit the release of algorithms that improve social welfare ex-post. Once the developer incurs the costs to develop an algorithm with an innovation level that is not socially optimal, the regulator has an incentive to release the algorithm if uncertainty about external effects is low; otherwise, conduct beta testing and condition the release of the algorithm in the second period on the test results. This approach yields higher welfare ex-post (after ℓ has been chosen) than simply banning the algorithm's release.

4.3.2 The optimal sequential testing and approval policy

We now consider a setting where the regulator has the same instruments as in Section 4.3.1 but has no commitment. The timing is as follows: (1) the developer chooses ℓ, (2) given ℓ, the regulator decides whether to allow immediate release or require beta testing, (3) nature draws the results of the beta test, generating posteriors, and (4) the regulator decides whether to allow the release of the algorithm in the second period.

The optimal sequential policy solves the following problem: for each ℓ

$$\mathcal{W}^{\text{reg}}(\ell) = \max_{\mu_1} \left\{ (1-\beta) \left(u(\ell) - \sigma^2_{s,i}(\ell) - (N+1)\, \sigma^2_{s,e}(\ell) \right) \mu_1 \right.$$
$$\left. \beta \mathbb{E}^s_1 \left\{ \max_{\mu_2} \left\{ \left(u(\ell) - \mathbb{E}^s_2 \left[\phi^2_i(\ell) \right] - (N+1)\, \mathbb{E}^s_2 \left[\phi^2_e(\ell) \right] \right) \mu_2 \right\} \right\} \right\} - f(\ell)$$

The regulator mandates beta testing if

$$\frac{u(\ell) - \sigma_{s,i}^2(\ell)}{N+1} < \sigma_{s,e}^2(\ell).$$

Otherwise, the algorithm is released. Given the test results, the regulator approves the release of the algorithm if

$$\frac{u(\ell) - [\hat{\phi}_{s,i}^2 + \hat{\sigma}_{s,i}^2(\ell)]}{N+1} \geq \hat{\phi}_{s,e}^2 + \hat{\sigma}_{s,e}^2(\ell),$$

and they forbid the algorithm's release, $\mu_2 = 0$, otherwise. So, for each ℓ, testing and release coincide with those of the social optimum.

We now show that when the regulator controls beta testing and release, the developer can have incentives to develop algorithms that are too risky from a social perspective, i.e., they choose a level of innovation ℓ that exceeds the socially optimal level. To establish this result, let $\varsigma_{s,x}^2(\ell)$ denote the regulator's ex-ante uncertainty about the AI's internal effects ($x = i$) or external effects ($x = e$) in period two:

$$\varsigma_{s,x}^2(\ell) \equiv \mathbb{E}_1^s \left(\mathbb{E}_2^s[\phi_x(\ell)^2] \mu_2^*(\ell) \right), \tag{3}$$

where $\mu_2^*(\ell)$ denotes the random variable which is equal to N if the regulator allows the AI to be commercialized in period two and equal to zero otherwise. We define $\varsigma_{d,x}^2(\ell)$ anologously according to the expectations of the developer.

Proposition 2. *Suppose that (i) the regulator implements the sequentially optimal testing and approval policy and (ii) both the first-best solution and the regulated equilibrium feature beta testing in period one.*

- *Suppose that beliefs are homogeneous. If $\varsigma_{s,e}^2(\ell)$ is increasing in ℓ, then the developer chooses a larger innovation level than the first best.*

- *Suppose that beliefs are heterogeneous. If $\varsigma_{s,e}^2(\ell)$ is increasing in ℓ, ex-ante uncertainty features decreasing differences*

$$\varsigma_{d,x}^2\left(\ell'\right) - \varsigma_{d,x}^2\left(\ell\right) \leq \varsigma_{s,x}^2\left(\ell'\right) - \varsigma_{s,x}^2\left(\ell\right),$$

and ex-ante expected revenue features increasing differences

$$\mathbb{E}_1^d[u(\ell')\mu_2^*(\ell')] - \mathbb{E}_1^d[u(\ell)\mu_2^*(\ell)] \geq \mathbb{E}_1^s[u(\ell')\mu_2^*(\ell')] - \mathbb{E}_1^s[u(\ell)\mu_2^*(\ell)],$$

then the developer chooses a larger innovation level than the first best.

In sum, developers can still have an incentive to take excessive risks by over-investing in AI, choosing innovation levels in excess of what is socially optimal. Although the policy described above does not guarantee the efficient use of resources, it allows society to harness the potential benefits of AI while mitigating its negative impacts.

This policy parallels the regulations currently governing the pharmaceutical industry in many countries. Pharmaceutical drugs undergo testing to assess their efficacy and side effects, with approval granted only if the expected benefits outweigh their expected costs. Similarly, AI algorithms are evaluated for their broader societal impacts and are approved only if their expected benefits exceed expected costs.

The private sector currently dominates the development of large AI models. According to Rahman et al. (2024), most large-scale AI algorithms have been developed by industry (71), with a smaller number resulting from industry-academia collaborations (6) and only a few created by academic (2) and government (2) institutions. To implement testing and approval policies like the one just discussed, governments must close the gap with the private sector, which will require significant public investment in computational infrastructure and expertise.

5 A model with internalities

In this section, we consider a model that incorporates deviations from rational behavior, known as internalities. These deviations lead households to make decisions that are not in their self-interest because of misinformation, self-control issues, cog-

nitive biases, or time inconsistency problems, all of which can be exploited by AI algorithms.

5.1 Unregulated equilibrium

Household's problem In Section 3, we assume that households take the expected welfare reduction caused by internal effects, $\mathbb{E}_t(i_t^2)$, into account when deciding whether to use the algorithm. Here, we consider the case where households disregard these internal effects due to behavioral biases when making their purchase decision.

We formalize this idea by assuming that $\mathcal{U}_j$, defined in equation (2), 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] Lifetime decision utility takes the form:

$$\mathcal{U}_j^b = (1-\beta)v_{j,1}^b + \beta \mathbb{E}_1(v_{j,2}^b),$$

where decision momentary utility is

$$v_{j,t}^b = y_t + [u(\ell) - p_t] \times \mathcal{I}_{j,t} - \mathbb{E}(e_t^2). \tag{4}$$

The household decides whether to purchase the AI algorithm to maximize $\mathcal{U}_j^b$. The resulting decision rule is to buy the algorithm whenever $p_t \leq u(\ell)$. Recall that in the absence of behavioral biases, the decision rule is to buy the algorithm when $p_t \leq u(\ell) - \mathbb{E}_t[\phi_i(\ell)^2]$.

We assume that the developer is immune to the algorithm's internal effects, either because it does not use the algorithm or is more sophisticated than the households.[4]

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

[4] Extending our analysis to the case where the algorithm's internal effects also affect the developer is straightforward. Such an extension would not significantly alter our findings.

What are the key differences between this model and our benchmark model? Because households ignore expected negative internal effects on utility, the developer can charge them a higher price: $p_t = u(\ell)$ instead of $p_t = u(\ell) - \mathbb{E}_t[\phi_i(\ell)^2]$.

Internalities widen the gap between the unregulated equilibrium and the social optimum. In period one, the developer beta tests the algorithm when $\sigma_e^2(\ell) > u(\ell)$ and releases the algorithm otherwise. In contrast, the planner has a lower threshold for the level of uncertainty required for beta testing. It is socially optimal to beta test whenever $\sigma_e^2(\ell) > [u(\ell) - \sigma_i^2(\ell)]/(N+1)$.

In period two, the developer withdraws the algorithm only when $\hat{\phi}_e^2 + \sigma_e^2(\ell) > u(\ell)$. The planner uses a lower uncertainty threshold for withdrawal. It is socially optimal to withdraw the algorithm whenever $\hat{\phi}_e^2 + \sigma_e^2(\ell) > [u(\ell) - \hat{\phi}_i^2 - \sigma_i^2(\ell)]/(N+1)$.

In the model with externalities, the developer overlooks the external impacts on the broader population but 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 with respect to internalities because the developer is not personally affected by internalities and the price does not reflect the internal effects experienced by households.

We can design ex-ante Pigouvian taxes that align private and social incentives. But, these taxes are even more complex than the ones discussed in Section 4.

The optimal time-consistent combination of beta testing and regulatory approval is the one described in subsection 4.3.2 and summarized in Table 1. However, this policy is more challenging to implement in the current setting because it requires regulators to consider external and internal effects. In subsection 4.3.2, households account for the internal effects in their purchasing decisions, eliminating the need for regulators to consider these effects.

6 Conclusion

In this paper, we study how to regulate AI, taking into account two key aspects of the algorithms currently being developed: significant uncertainty regarding their potential social costs and widespread disagreement about the likelihood of these costs.

Our analysis yields two key insights. First, the complexity of Pigouvian taxes needed to align private and social incentives renders them impractical. Second, a combination of beta testing and regulatory approval can mitigate AI's risks while still harnessing its benefits.

What are the implications of our model for the efficacy of the regulatory proposals currently being discussed in the U.S. and the European Union? Simply banning the development of algorithms that pose a high risk of negative externalities is insufficient to achieve the social optimum because the unregulated equilibrium diverges from the social optimum at intermediate levels of uncertainty (see Table 1). Holding developers liable for external effects can lead to excessive risk-taking for two reasons. First, liability is typically capped in practice, which allows developers to overlook negative externalities that exceed their liability limits. Second, developers may have a more optimistic outlook about external effects than regulators, leading them to take on more risk than is socially optimal, even with full liability in place.

We discuss optimal AI regulation in a single-country setting. However, international cooperation is generally required when algorithm use in one country imposes external effects on other countries.

Implementing beta testing and regulatory approval and coordinating these policies worldwide requires substantial public investment in computational resources and expertise. It is a formidable task but also an urgent one. As Isaac Asimov observed, "The saddest aspect of life right now is that science gathers knowledge faster than society gathers wisdom."

References

ACEMOGLU, D. (2021): "Harms of AI," in *The Oxford Handbook of AI Governance*, Oxford University Press.

ACEMOGLU, D. AND T. LENSMAN (2023): "Regulating Transformative Technologies," Tech. rep., National Bureau of Economic Research.

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.

AMODEI, D., C. OLAH, J. STEINHARDT, P. CHRISTIANO, J. SCHULMAN, AND D. MANÉ (2016): "Concrete problems in AI safety," *arXiv preprint arXiv:1606.06565*.

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.

BLATTNER, L., S. NELSON, AND J. SPIESS (2021): "Unpacking the black box: Regulating algorithmic decisions," *arXiv preprint arXiv:2110.03443*.

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.

CALLANDER, S. AND H. LI (2024): "Regulating an Innovative Industry," Working paper.

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.

GANS, J. S. (2024): "How Learning About Harms Impacts the Optimal Rate of Artificial Intelligence Adoption," Working Paper 32105, National Bureau of Economic Research.

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

HART, R. (2024): "AI models like ChatGPT won't reach human intelligence, Meta's AI chief says," *Forbes*.

HEAVEN, W. D. (2023): "Geoffrey Hinton tells us why he's now scared of the tech he helped build," *MIT Technology Review*.

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.

IDE, E. AND E. TALAMAS (2023): "Artificial Intelligence in the Knowledge Economy," *arXiv preprint arXiv:2312.05481*.

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.

JUMPER, J., R. EVANS, A. PRITZEL, T. GREEN, M. FIGURNOV, O. RONNEBERGER, K. TUNYASUVUNAKOOL, R. BATES, A. ŽÍDEK, A. POTAPENKO, ET AL. (2021): "Highly accurate protein structure prediction with AlphaFold," *nature*, 596, 583–589.

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.

MARTINEZ, J. (2021): "Putty-clay automation," *DP16022*.

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

RAHMAN, R., D. OWEN, AND J. YOU (2024): "Tracking Large-Scale AI Models," .

RUSSELL, S., D. DEWEY, AND M. TEGMARK (2015): "Research priorities for robust and beneficial artificial intelligence," *AI Magazine*, 36, 105–114.

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.

WIENER, N. (1960): "Some Moral and Technical Consequences of Automation," *Science*, 131, 1355–58.

A Example where social optimum has a higher level of innovation than the unregulated equilibrium

In this appendix, we provide an example in which, by being more cautious in beta testing and algorithm release, the social planner chooses a higher level of innovation than the developer.

The numerical example is as follows. Suppose that $u(\ell) = \sqrt{\ell}$, $f(\ell) = \chi\ell^2/2$ with $\chi = 10$, and that there are no internalities $\phi_i(\ell) = 0$. In addition, assume that $\beta = 0.7$ and that $\phi_e(\ell)$ is such that

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

We set $\varphi = 1.0079$ and $\alpha = 0.1$. In this case:

$$\sigma_e^2(\ell) = (1-\alpha)\,\varphi^2\ell^4.$$

We assume that beta testing fully reveals the external effect at the end of period one.

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

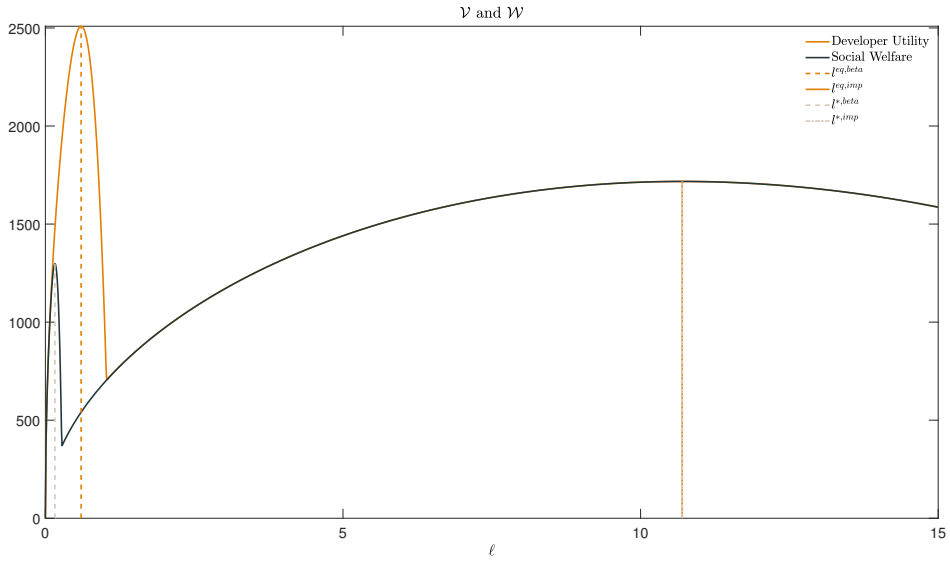

Figure 1: Example where social optimum has higher level of innovation than the unregulated equilibrium

B Proof of Proposition 2

Let $\mathcal{O}(\ell, a)$ denote the objective function of the developer for a given ℓ if $a = 1$, given the testing and approval policy, or the objective function of the first-best planner if $a = 0$.

$$\begin{aligned}
\mathcal{O}(\ell,0) \equiv & -f(\ell) + (1-\beta)\left\{u(\ell) - \sigma^2_{s,i}(\ell) - (N+1)\,\sigma^2_{s,e}(\ell)\right\}\mu^*_1(\ell) \\
& +\beta\left\{\mathbb{E}^s_1\left[u(\ell)\mu^*_2(\ell)\right] - \varsigma^2_{s,i}(\ell) - (N+1)\varsigma^2_{s,e}(\ell)\right\} \\
\mathcal{O}(\ell,1) \equiv & -f(\ell) + (1-\beta)\left\{u(\ell) - \sigma^2_{d,i}(\ell) - \sigma^2_{d,e}(\ell)\right\}\mu^*_1(\ell) \\
& +\beta\left\{\mathbb{E}^d_1\left[u(\ell)\mu^*_2(\ell)\right] - \varsigma^2_{d,i}(\ell) - \varsigma^2_{d,e}(\ell)\right\},
\end{aligned}$$

where $\mu^*_t(\ell)$ denotes the optimal testing and implementation strategy as defined by the regulator (which coincides with the first-bets one). The first-best level of innovation solves $\max_\ell \mathcal{O}(\ell, 0)$, while the optimal level of innovation for the developer

solves $\max_\ell \mathcal{O}(\ell, 1)$. Our proof strategy is to show that $\mathcal{O}(\ell, a)$ satisfies the single-crossing property and then apply the monotone comparative statics results in Milgrom and Shannon (1994) to establish that $\ell^*(1) \geq \ell^*(0)$.[5]

Since $\ell^*(1)$ and $\ell^*(0)$ are such that beta testing is needed, we focus only on choices made in the set $(\overline{\ell}, \infty)$ where $\overline{\ell}$ is such that $u(\overline{\ell}) - \sigma^2_{s,i}(\overline{\ell}) - (N+1)\sigma^2_{s,e}(\overline{\ell}) = 0$, i.e., for all $\ell > \overline{\ell}$, $u(\ell) - \sigma^2_{s,i}(\ell) - (N+1)\sigma^2_{s,e}(\ell) < 0$.

Take $\ell' > \ell \in (\overline{\ell}, \infty)$, we show that

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

Note that $\mathcal{O}(\ell', 0) \geq \mathcal{O}(\ell, 0)$ implies that

$$\begin{aligned}
&-f(\ell') + \beta\left\{\mathbb{E}^s_1\left[u(\ell')\mu^*_2(\ell')\right] - \varsigma^2_{s,i}(\ell') - (N+1)\varsigma^2_{s,e}(\ell')\right\} \\
\geq &-f(\ell) + \beta\left\{\mathbb{E}^s_1\left[u(\ell)\mu^*_2(\ell)\right] - \varsigma^2_{s,i}(\ell) - (N+1)\varsigma^2_{s,e}(\ell)\right\} \\
\Leftrightarrow &-f(\ell') + \beta\left\{\mathbb{E}^s_1\left[u(\ell')\mu^*_2(\ell')\right] - \varsigma^2_{s,i}(\ell') - (N+1)[\varsigma^2_{s,e}(\ell') - \varsigma^2_{s,e}(\ell)]\right\} \\
\geq &-f(\ell) + \beta\left\{\mathbb{E}^s_1\left[u(\ell)\mu^*_2(\ell)\right] - \varsigma^2_{s,i}(\ell)\right\}
\end{aligned}$$

since $\varsigma^2_{s,e}(\ell)$ is increasing in ℓ then $\varsigma^2_{s,e}(\ell') - \varsigma^2_{s,e}(\ell) > 0$, which implies that

$$\begin{aligned}
\Rightarrow &-f(\ell') + \beta\left\{\mathbb{E}^s_1\left[u(\ell')\mu^*_2(\ell')\right] - \varsigma^2_{s,i}(\ell') - [\varsigma^2_{s,e}(\ell') - \varsigma^2_{s,e}(\ell)]\right\} \\
> &-f(\ell') + \beta\left\{\mathbb{E}^s_1\left[u(\ell')\mu^*_2(\ell')\right] - \varsigma^2_{s,i}(\ell') - (N+1)[\varsigma^2_{s,e}(\ell') - \varsigma^2_{s,e}(\ell)]\right\} \\
\geq &-f(\ell) + \beta\left\{\mathbb{E}^s_1\left[u(\ell)\mu^*_2(\ell)\right] - \varsigma^2_{s,i}(\ell)\right\}
\end{aligned}$$

Taking the first and last expressions in that inequality sequence, we obtain

$$\begin{aligned}
&-f(\ell') + \beta\left\{\mathbb{E}^s_1\left[u(\ell')\mu^*_2(\ell')\right] - \varsigma^2_{s,i}(\ell') - [\varsigma^2_{s,e}(\ell') - \varsigma^2_{s,e}(\ell)]\right\} \\
&> -f(\ell) + \beta\left\{\mathbb{E}^s_1\left[u(\ell)\mu^*_2(\ell)\right] - \varsigma^2_{s,i}(\ell)\right\} \\
\Leftrightarrow &-f(\ell') + \beta\left\{\mathbb{E}^s_1\left[u(\ell')\mu^*_2(\ell')\right] - \varsigma^2_{s,i}(\ell') - \varsigma^2_{s,e}(\ell')\right\} \\
&> -f(\ell) + \beta\left\{\mathbb{E}^s_1\left[u(\ell)\mu^*_2(\ell)\right] - \varsigma^2_{s,i}(\ell) - \varsigma^2_{s,e}(\ell)\right\},
\end{aligned}$$

[5] For simplicity, we assume that the maximizer set is a singleton.

which implies that $\mathcal{O}(\ell', 1) > \mathcal{O}(\ell, 1)$ if beliefs are homogeneous.

With heterogeneous beliefs, we can equivalently write:

$$-f\left(\ell'\right) + \beta\left\{\mathbb{E}_1^s\left[u(\ell')\mu_2^*(\ell')\right] - \mathbb{E}_1^s\left[u(\ell)\mu_2^*(\ell)\right] - \left[\varsigma_{s,i}^2(\ell') - \varsigma_{s,i}^2(\ell)\right] - \left[\varsigma_{s,e}^2(\ell') - \varsigma_{s,e}^2(\ell)\right]\right\} > -f(\ell).$$

Assuming that

$$\begin{aligned}
&\mathbb{E}_1^d\left[u(\ell')\mu_2^*(\ell')\right] - \mathbb{E}_1^d\left[u(\ell)\mu_2^*(\ell)\right] \geq \mathbb{E}_1^s\left[u(\ell')\mu_2^*(\ell')\right] - \mathbb{E}_1^s\left[u(\ell)\mu_2^*(\ell)\right] \\
&\varsigma_{d,x}^2(\ell') - \varsigma_{d,x}^2(\ell) \leq \varsigma_{s,i}^2(\ell') - \varsigma_{s,i}^2(\ell),
\end{aligned}$$

we find that the above expression implies

$$\begin{aligned}
&-f\left(\ell'\right) + \beta\left\{\mathbb{E}_1^d\left[u(\ell')\mu_2^*(\ell')\right] - \mathbb{E}_1^d\left[u(\ell)\mu_2^*(\ell)\right] - \left[\varsigma_{d,i}^2(\ell') - \varsigma_{d,i}^2(\ell)\right] - \left[\varsigma_{d,e}^2(\ell') - \varsigma_{d,e}^2(\ell)\right]\right\} \\
&\geq -f\left(\ell'\right) + \beta\left\{\mathbb{E}_1^s\left[u(\ell')\mu_2^*(\ell')\right] - \mathbb{E}_1^s\left[u(\ell)\mu_2^*(\ell)\right] - \left[\varsigma_{s,i}^2(\ell') - \varsigma_{s,i}^2(\ell)\right] - \left[\varsigma_{s,e}^2(\ell') - \varsigma_{s,e}^2(\ell)\right]\right\} \\
&> -f(\ell).
\end{aligned}$$

We can equivalently represent this expression as

$$\begin{aligned}
&-f\left(\ell'\right) + \beta\left\{\mathbb{E}_1^d\left[u(\ell')\mu_2^*(\ell')\right] - \varsigma_{d,i}^2(\ell') - \varsigma_{d,e}^2(\ell')\right\} \\
&> -f(\ell) + \beta\left\{\mathbb{E}_1^d\left[u(\ell)\mu_2^*(\ell)\right] - \varsigma_{d,i}^2(\ell) - \varsigma_{d,e}^2(\ell)\right\} \\
\Leftrightarrow &\mathcal{O}(\ell', 1) > \mathcal{O}(\ell, 1).
\end{aligned}$$

These results establish that $\mathcal{O}(\ell, a)$ satisfies the strict single-crossing property. Following Milgrom and Shannon (1994), we now establish that this property implies that the maximizer is greater when $a = 1$ than under $a = 0$. To establish a contradiction, suppose that $\ell^*(0) > \ell^*(1)$. Then, since $\ell^*(0)$ is a maximizer, it satisfies

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

By the single-crossing property, the previous expression implies that

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

which contradicts the assumption that $\ell^*(1)$ maximizes $\mathcal{O}(\ell, 1)$.

Online Appendix
Regulating Artificial Intelligence

A Model where beta testing outcomes depend on sample size

This section considers a version of the model where the outcome of beta testing depends on the number of people who participate in the test. To evaluate external misalignments, the developer can test the algorithm in a sample of μ_1 users in the first period. Based on the outcomes of this test, the developer can decide whether to release the algorithm in the second period. Despite the differences in the beta testing process, our qualitative results are essentially unchanged in this generalized framework.

We assume that the probability that the beta test generates information depends on the number of individuals involved in the test. The beta test is successful with probability $\pi(\mu_1)$, upon which the expectations are updated as in the baseline model. With probability $1 - \pi(\mu_1)$, the test generates no information. We define an indicator function $\mathcal{B}$ that takes the value one if the test is successful, and zero otherwise.[6] For concreteness, we assume the following functional form $\pi(\mu_1) = (\mu_1/\kappa)^\alpha$ if $\mu_1 \leq \kappa$ and $\pi(\mu_1) = 1$ otherwise. Here, $\kappa \leq N$ denotes the minimal number of participants required to learn $\phi_e(\ell)$ with certainty. If $\kappa = N$, we only learn $\phi(\ell)$ with certainty by releasing the software to the whole population.

The parameter α determines the test's effectiveness. As $\alpha \to 0$, then $\pi(\mu_1) \to 1$ if $\mu_1 > 0$ and $\pi(\mu_1) = 0$ if $\mu_1 = 0$. In this limiting case, minimal beta testing generates information with certainty. As $\alpha \to \infty$, $\pi(\mu_1) = 0$ if $\mu_1 < \kappa$ and $\pi(\mu_1) = 1$ if $\mu_1 = \kappa$. In this case, testing reveals $\phi(\ell)$ with certainty only if the entire population

[6] The baseline model is a special case of this model in which $\pi(0) = 0$ and $\pi(\mu_1) = 1$ if $\mu_1 > 0$.

participates in the test.

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

A.1 Unregulated equilibrium

Household problem The household problem is the same as that in the baseline model. The household is willing to purchase an AI license in period t if the private benefits exceed the price of the algorithm

$$u(\ell) - \mathbb{E}_t[\phi_i(\ell)^2] \geq p_t.$$

AI developer's problem The developer's problem is analogous to the baseline model, modified to include the effect of μ_1 on the likelihood of obtaining information about external effects.

The solution to the time two problem is the same. The developer chooses the price $p_2 = u(\ell) - \mathbb{E}_2[\phi_i(\ell)^2]$ and releases the algorithm if $u(\ell) - \mathbb{E}_2[\phi_i(\ell)^2] \geq \mathbb{E}_2[\phi_e(\ell)^2]$.

Let $\mathbb{E}_1[\mathcal{V}_2^*(\ell)]$ denote the expected value of the developer's period two utility computed at the beginning of period one. Then,

$$\frac{d\mathbb{E}_1[\mathcal{V}_2^*(\ell)]}{d\mu_1} = \pi'(\mu_1) \times \left(\mathbb{E}_1 \left[\max \left\{ u(\ell) - \sum_x \mathbb{E}_2[\phi_x(\ell)^2] N, 0 \right\} \right] - \max \left\{ u(\ell) - \sum_x \sigma_x^2(\ell) N, 0 \right\} \right) \geq 0$$

which establishes the analogue of Lemma 1:

Lemma 3. *The developer's expected utility in the second period is increasing in μ_1.*

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

$$\mathcal{V} = (1-\beta) \left(\begin{cases} \mu_1 p_1 - \sigma_e^2(\ell)\mu_1, & \text{if } p_1 \leq u(\ell) - \sigma_i^2(\ell) \\ 0, & \text{if } p_1 > u(\ell) - \sigma_i^2(\ell) \end{cases} \right) + \beta \mathbb{E}_1[\mathcal{V}_2^*(\ell)] - f(\ell).$$

The optimal price for the developer is $p_1 = u(\ell) - \sigma_i^2(\ell)$.

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

If $u(\ell) - \sigma_i^2(\ell) - \sigma_e^2(\ell) \geq 0$, then releasing the algorithm generates expected gains in period one and increases expected utility in period two. Therefore, it is optimal to release the algorithm to the whole population, $\mu_1 = N$.

Instead, if $u(\ell) - \sum_x \sigma_x^2(\ell) < 0$, releasing the algorithm to the whole population may not be optimal. As long as $\alpha \leq 1$, the developer's utility is increasing in the neighborhood of $\mu_1 = 0$, so the optimal solution features $\mu_1 > 0$. The intuition for this result is that the benefits from learning increase sufficiently fast with μ_1 to offset the costs of testing, which are given by $[u(\ell) - \sum_x \sigma_x^2(\ell)]\mu_1^2$. Instead, if $\alpha > 1$, the utility is convex, so the solution is either $\mu_1 = 0$ or $\mu_1 = \kappa$ depending on parameters. The intuition for this result is that when α is large, beta tests on small samples are unlikely to generate information about external effects. The developer then adopts an "all-or-nothing" approach: they either do not release the algorithm or release it to the whole population. Proposition 3 summarizes the optimal release in periods one and two from the developer's point of view. In describing the solution, it is useful to define the information benefit-cost ratio, $\Lambda^d(\ell)$:

$$\Lambda^d(\ell) \equiv \frac{\beta}{1-\beta} \frac{\mathbb{E}\left[\max\left\{u(\ell) - \sum_x \mathbb{E}_2[\phi_x(\ell)^2], 0\right\}\right]}{\sum_x \sigma_x^2(\ell) - u(\ell)}.$$

This variable compares the expected benefits of increasing the probability of learning the external effects of the AI algorithm, $\beta\mathbb{E}\left[\max\left\{u(\ell) - \sum_x \mathbb{E}_2[\phi_x(\ell)^2], 0\right\}\right]$, to the immediate costs of selling the AI algorithm to an additional person today, $(1-\beta)[\sum_x \sigma_x^2(\ell) - u(\ell)]$.

Proposition 3 (Uncertainty, beta testing, and algorithm release)**.** *In an unregulated equilibrium, the number of user licenses offered by the developer in the first period depends*

on the level of uncertainty, the effectiveness of beta testing, and the information benefit-cost ratio. The solution is as follows:

1. *The developer always foregoes beta testing and releases the AI algorithm to the entire population in the first period ($\mu_1 = N$) when uncertainty about external effects is low $\sigma_e^2(\ell) \leq u(\ell) - \sigma_i^2(\ell)$.*

2. *If uncertainty about external effects is relatively high $\sigma_e^2(\ell) > u(\ell) - \sigma_i^2(\ell)$, then:*

   - *If $\alpha \leq 1$, beta testing is sufficiently effective. The developer beta tests the algorithm on*
   $$\mu_1 = \min\left\{\left[\alpha\Lambda^d(\ell)\frac{N}{\kappa}\right]^{\frac{1}{1-\alpha}}, 1\right\}\kappa. \tag{5}$$
   - *If $\alpha > 1$. Then the developer chooses the largest beta-test sample $\mu_1 = \kappa$ if the information benefit-cost ratio is larger than the ratio of maximum test size to total population, $\Lambda^d(\ell) \geq \kappa/N$. If the information benefit-cost ratio is small, the developer neither releases nor beta tests the algorithm, $\mu_1 = 0$.*

A.2 The first-best solution (planner's problem)

The planner's problem is analogous to the baseline model. If the AI algorithm is implemented in the first period, the planner learns its externality with probability $\pi(\mu_1)$. In the second period, the planner decides whether to make the AI algorithm available and how many licenses to offer.

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.

The solution to the period-two problem is the same. The planner chooses to allow licenses at if $u(\ell) - \mathbb{E}_2[\phi_i(\ell)^2] \geq (N+1)\mathbb{E}_2[\phi_e(\ell)^2]$. Let $\mathbb{E}_1[\mathcal{W}_2^*(\ell)]$ denote the expected value of the developer's period-two utility evaluated in the beginning of period one. Then note that

$$\frac{d\mathbb{E}_1[\mathcal{W}_2^*(\ell)]}{d\mu_1} = \pi'(\mu_1) \times \mathbb{E}_1\left[\max\left\{u(\ell) - \mathbb{E}_2[\phi_i(\ell)^2] - (N+1)\mathbb{E}_2[\phi_e(\ell)^2], 0\right\} N\right]$$
$$- \pi'(\mu_1) \times \max\left\{u(\ell) - \sigma_i^2(\ell) - (N+1)\sigma_e^2(\ell), 0\right\} N \geq 0$$

which establishes the analog of Lemma 2:

Lemma 4. *Expected social welfare in the second period is increasing in μ_1.*

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

$$\mathcal{W} = (1-\beta)\left(\begin{cases} Ny_1 + \mu_1[u(\ell) - \sigma_i^2(\ell) - (N+1)\sigma_e^2(\ell)]\mu_1, & \text{if } p_1 \leq u(\ell) - \sigma_i^2(\ell) \\ 0, & \text{if } p_1 > u(\ell) - \sigma_i^2(\ell) \end{cases}\right) + \beta\mathbb{E}_1[\mathcal{W}_2^*(\ell)] - f(\ell).$$

From a static perspective, it is still optimal to set $\mu_1 = N$ if $u(\ell) - \sigma_i^2(\ell) - (N+1)\sigma_e^2(\ell) \geq 0$ and $\mu_1 = 0$ if $u(\ell) - \sigma_i^2(\ell) - (N+1)\sigma_e^2(\ell) < 0$. However, experimenting in the first period, $\mu_1 > 0$ creates value by generating information that the developer can use in the second period.

If $u(\ell) - \sigma_i^2(\ell) - (N+1)\sigma_e^2(\ell) \geq 0$, then releasing the algorithm generates expected gains in period one and increases expected utility in period two. Therefore, it is optimal to release the algorithm to the whole population, $\mu_1 = N$.

Instead, if $u(\ell) - \sigma_i^2(\ell) - (N+1)\sigma_e^2(\ell) < 0$, releasing the algorithm to the whole population may not be optimal. As long as $\alpha \leq 1$, the developer's utility is increasing in the neighborhood of $\mu_1 = 0$, so the optimal solution features $\mu_1 > 0$. If $\alpha > 1$, the utility is convex, so the solution is either $\mu_1 = 0$ or $\mu_1 = \kappa$ depending on parameters. The intuition for these results is similar to that described in the discussion of the developer's problem.

Proposition 3 summarizes the optimal release policy in periods one and two from the developer's point of view. In describing the solution, it is useful to define the information benefit-cost ratio, $\Lambda^s(\ell)$:

$$\Lambda^s(\ell) \equiv \frac{\beta}{1-\beta}\frac{\mathbb{E}\left[\max\left\{u(\ell) - \mathbb{E}_2[\phi_i(\ell)^2] - (N+1)\mathbb{E}_2[\phi_e(\ell)^2], 0\right\}\right]}{(N+1)\sigma_e^2(\ell) + \sigma_i^2(\ell) - u(\ell)}.$$

This variable compares the expected benefits of increasing the probability of learning the external effects of the AI algorithm, $\beta\mathbb{E}\left[\max\left\{u\left(\ell\right)-\mathbb{E}_2[\phi_i\left(\ell\right)^2]-(N+1)\mathbb{E}_2[\phi_e\left(\ell\right)^2],0\right\}\right]$, to the immediate costs of selling the AI algorithm to an additional person today, $(1-\beta)[(N+1)\sigma_i^2\left(\ell\right)+\sigma_i^2\left(\ell\right)-u\left(\ell\right)]$.

Proposition 4 (Uncertainty, beta testing, and algorithm release)**.** *In the first best, the number of user licenses offered in the first period depends on the level of uncertainty, the effectiveness of beta testing, and the information benefit-cost ratio. The solution is as follows:*

1. *The planner always foregoes beta testing and releases the AI algorithm to the entire population in the first period ($\mu_1 = N$) when uncertainty is low $\sigma_e^2(\ell) \leq \frac{u(\ell)-\sigma_i^2(\ell)}{N+1}$.*
2. *If uncertainty is relatively high $\sigma_e^2(\ell) > \frac{u(\ell)-\sigma_i^2(\ell)}{N+1}$, then:*
   - *If $\alpha \leq 1$, beta testing is sufficiently effective. The developer beta tests the algorithm on*
   $$\mu_1 = \min\left\{\left[\alpha\Lambda^s(\ell)\frac{N}{\kappa}\right]^{\frac{1}{1-\alpha}},1\right\}\kappa. \tag{6}$$
   - *If $\alpha > 1$. Then the developer chooses the largest beta-test sample $\mu_1 = \kappa$ if the information benefit-cost ratio is larger than one, $\Lambda^s(\ell) \geq \kappa/N$. If the information benefit-cost ratio is small, the developer neither releases nor beta tests the algorithm, $\mu_1 = 0$.*

The planner is more conservative than the developer when deciding whether to conduct beta tests instead of releasing the algorithm to the whole population. There are AI innovation levels for which the developer prefers an immediate release to the general public, while the planner opts for beta testing. Moreover, the planner tends to favor more cautious beta tests that involve smaller sample sizes.

When $\alpha \leq 1$, beta testing is relatively effective. In this case, when uncertainty is sufficiently high, the developer and planner agree to beta-test the algorithm. However, they disagree on the acceptable risk in that beta test–the risk level increases with

the number of testers involved. Since the planner has a smaller information benefit-cost ratio than the developer, the planner favors smaller sample sizes, $\mu_1^* \leq \mu_1^e$.

When $\alpha > 1$, beta testing is relatively ineffective. In this case, the planner and developer favor "all-or-nothing" strategies in which the algorithm is either fully released in period one or not. However, the planner remains more cautious, requiring a higher information benefit-cost ratio than the developer for a full market release.

Upon learning some of the external effects of the AI algorithm in the initial period, there are scenarios where the developer sees continued commercialization in the second period as privately beneficial, while the planner opts to pull the algorithm from the market. If the algorithm's external consequences are not learned, the planner is also more cautious than the developer regarding the release of the AI in the second period. There are medium levels of uncertainty where the developer is willing to proceed with a full release, but the planner deems it too risky.

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.

B Regulating AI

Despite the differences between the two models, our results on Pigouvian taxes are unchanged. Ex-post full liability delivers the efficient allocation with homogeneous beliefs. However, it fails to do so when developers are protected with limited liability. Ex-ante liability, while immune to the limited liability concern, is very difficult to implement if developers and the regulator have different beliefs.

In terms of testing and approval policies, there continues to be a policy that ensures that the equilibrium delivers the efficient allocation. This policy requires the regulator to commit to rejecting any algorithm that does not feature the optimal innovation level, controlling the test size, and conditionally approving any algorithm

with the optimal innovation level. As before, this policy is not time-consistent because the regulator must commit to rejecting algorithms that might be welfare improving ex-post.

The optimal sequential testing and approval policy is now more complicated because beta testing is not certain to produce information in this generalized model. In period two, if the beta test was unsuccessful, then the regulator allows the developer to sell AI licenses if and only if

$$\sigma^2_{s,e} \leq \frac{u(\ell) - \sigma^2_{s,i}(\ell)}{N+1}.$$

If the beta test was successful, then the regulator conditionally approves the algorithm in the second period if and only if

$$\hat{\phi}^2_{s,e} + \hat{\sigma}^2_{s,e}(\ell) \leq \frac{u(\ell) - \hat{\phi}^2_{s,i} - \hat{\sigma}^2_{s,i}(\ell)}{N+1}.$$

In period one, the regulator allows free commercialization of the algorithm if $\sigma^2_{s,e} \leq \frac{u(\ell)-\sigma^2_{s,i}(\ell)}{N+1}$. Instead, if uncertainty is too high and $\alpha \geq 1$, the regulator forbids commercialization of the AI if $\Lambda^s(\ell) < \kappa/N$ and allows free commercialization of the AI otherwise. If uncertainty is too high and $\alpha < 1$, then the regulator mandates a size of the beta test which solves

$$\mu_1 = \min\left\{\left[\alpha\Lambda^s(\ell)\frac{N}{\kappa}\right]^{\frac{1}{1-\alpha}}, 1\right\}\kappa.$$

These conditions are the natural generalization of the baseline model to this more general setting.