Property Rights for Innovation:
A Structural Model of Patent Screening

Online Appendices

William Matcham and Mark Schankerman

Contents

A Additional Tables and Figures

Table A.1. Application Fighting Costs by Technology Area (Internal Estimation)

Parameter	Symbol	Estimate	S.E.
Chemical application fighting cost log-mean	$\mu_{f,\text{chem}}$	9.15	0.008
Chemical application fighting cost log-sigma	$\sigma_{f,\text{chem}}$	0.38	0.010
Electrical application fighting cost log-mean	$\mu_{f,\text{elec}}$	9.18	0.010
Electrical application fighting cost log-sigma	$\sigma_{f,\text{elec}}$	0.57	0.014
Mechanical application fighting cost log-mean	$\mu_{f,\text{mech}}$	9.02	0.008
Mechanical application fighting cost log-sigma	$\sigma_{f,\text{mech}}$	0.47	0.011

Notes: Standard errors are bootstrapped.

Table A.2. Applicant Fighting Costs by Technology Area (External Estimation)

Parameter	Symbol	Estimate
Simple amendment fighting cost log-mean	$\mu_{f,\text{amend,simp}}$	7.60
Simple amendment fighting cost log-sigma	$\sigma_{f,\text{amend,simp}}$	0.37
Chemical amendment fighting cost log-mean	$\mu_{f,\text{amend,chem}}$	8.13
Chemical amendment fighting cost log-sigma	$\sigma_{f,\text{amend,chem}}$	0.45
Electrical amendment fighting cost log-mean	$\mu_{f,\text{amend,elec}}$	8.07
Electrical amendment fighting cost log-sigma	$\sigma_{f,\text{amend,elec}}$	0.38
Mechanical amendment fighting cost log-mean	$\mu_{f,\text{amend,mech}}$	7.95
Mechanical amendment fighting cost log-sigma	$\sigma_{f,\text{amend,mech}}$	0.43
Issuance cost log-mean	$\mu_{f,\text{iss}}$	6.54
Issuance cost log-sigma	$\sigma_{f,\text{iss}}$	0.62
Maintenance cost log-mean	$\mu_{f,\text{main}}$	5.67
Maintenance cost log-sigma	$\sigma_{f,\text{main}}$	0.46

Table A.3. Robustness of Estimates

Parameter	**Symbol**	**Baseline**	1% $\boldsymbol{\tau}$	5% $\boldsymbol{\tau}$	$\boldsymbol{\beta = 0.99}$	**Definition of Seniority (GS13 + GS14)**
Junior intrinsic motivation log-mean	$\mu_{\theta,\mathrm{j}}$	3.92	3.96	3.96	3.90	4.16
Senior intrinsic motivation log-mean	$\mu_{\theta,\mathrm{s}}$	3.38	2.90	2.73	3.18	2.93
Intrinsic motivation log-sigma	σ_θ	0.77	0.82	0.79	0.90	0.99
Examiner delay cost log-mean	μ_π	0.19	0.16	0.18	0.49	0.12
Examiner delay cost log-sigma	σ_π	0.27	0.37	0.42	0.10	0.60
Error standard deviation	σ_ε	0.02	0.02	0.02	0.03	0.02
Initial returns log-mean	μ_v	10.55	10.59	10.88	10.07	10.28
Initial returns log-sigma	σ_v	1.32	1.13	1.61	2.94	0.57
Initial distance alpha	α_D	4.57	3.92	3.90	4.56	3.75
Initial distance beta	β_D	7.74	6.72	6.22	7.79	7.15
Narrowing probability	η	0.75	0.73	0.74	0.75	0.72
Application obsolescence probability	$P_{\omega,\mathrm{app}}$	0.14	0.13	0.13	0.12	0.14
Renewal obsolescence probability	$P_{\omega,\mathrm{renew}}$	0.04	0.04	0.04	0.04	0.04
Simple application fighting cost log-mean	$\mu_{f,\mathrm{simple}}$	8.53	8.43	8.56	8.60	8.53
Simple application fighting cost log-sigma	$\sigma_{f,\mathrm{simple}}$	0.87	0.97	0.79	0.74	0.95
SMM Objective		1.23	1.47	1.29	1.25	1.33

Notes: This table provides estimates of the model parameters across various model alternatives.

Figure A.1. Match of internal data and model moments

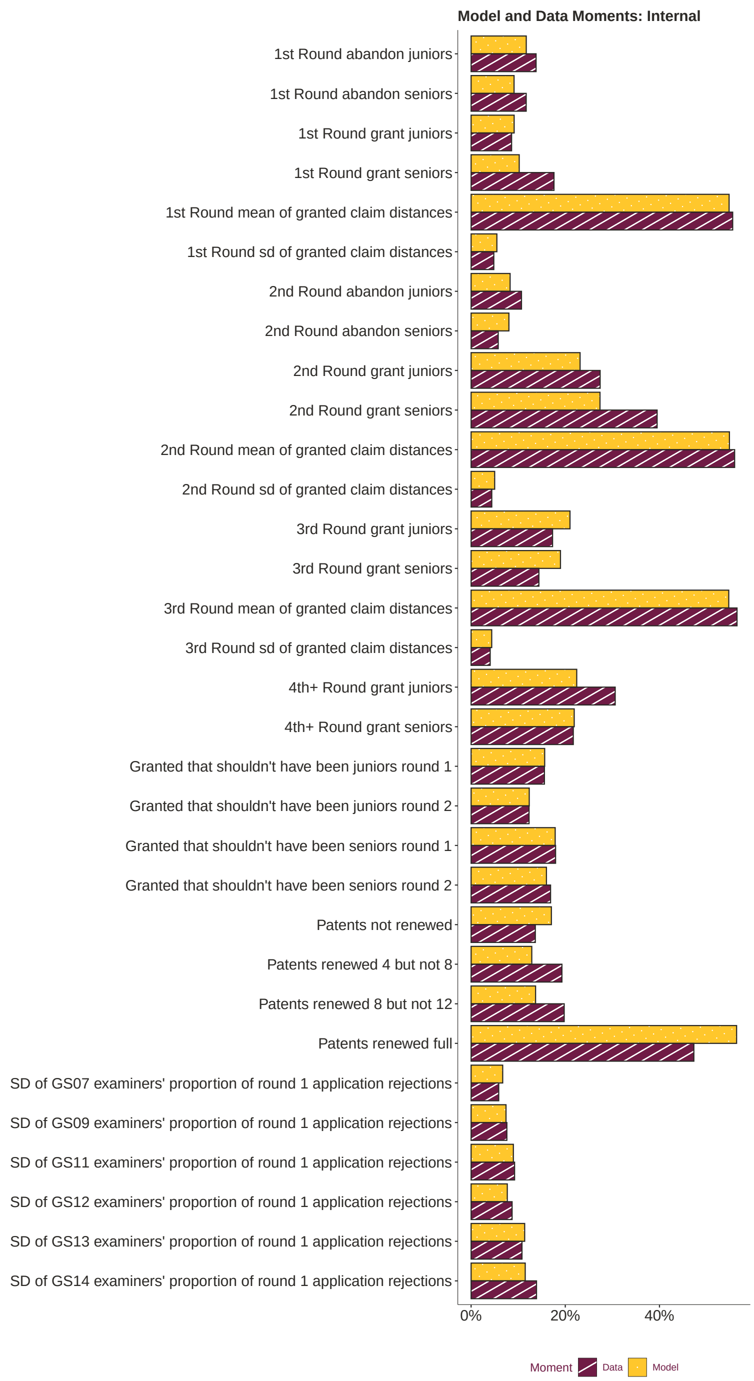
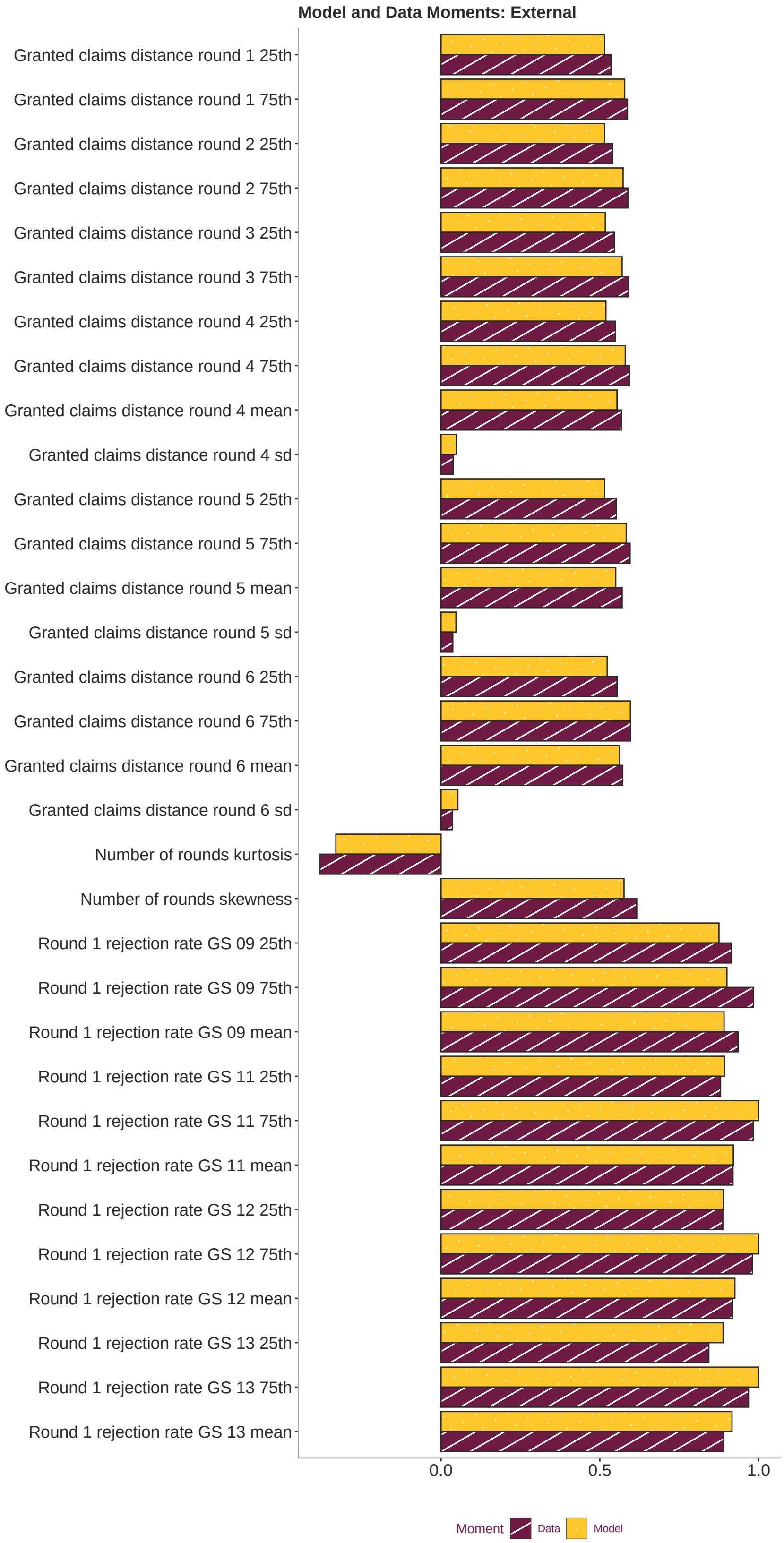


Figure A.2. Match of external data and model moments



B Data Sources

All links below last accessed 19 July 2023. If the links are broken, the documents are available upon request.

B.1 Publicly Available Datasets

1. Data on patent application and granted patent claim text are from the *U.S.PTO Patent Application Claims Full Text Dataset* and *U.S. PTO Patent Claims Full Text Dataset*. The data can be downloaded from https://www.uspto.gov/learning-and-resources/electronic-data-products/patent-claims-research-dataset

2. Data on the written communication between applicant and examiner, along with information on art units, examiners, and applicants are from the *Patent Examination (PatEx) Research Dataset*. The data can be downloaded from https://www.uspto.gov/ip-policy/economic-research/research-datasets/patent-examination-research-dataset-public-pair

3. Data on patent renewals are from the *U.S.PTO Maintenance Fee Events Dataset*. The data can be downloaded from https://developer.uspto.gov/product/patent-maintenance-fee-events-and-description-files

4. Data on patent rejections are from the *U.S.PTO Office Action Research Dataset*. The data can be downloaded from https://www.uspto.gov/ip-policy/economic-research/research-datasets/office-action-research-dataset-patents

5. Data on the seniority of examiners over time are from *Frakes and Wasserman (2019)*. The data can be downloaded from https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/ABE7VS

B.2 Data from Public Documents

6. Data on GDP deflators were obtained from the St. Louis Federal Reserve. See https://fred.stlouisfed.org/series/GDPDEF.

7. Data on the costs of patent litigation were collected from the 2005, 2011, and 2017 editions of the *American Intellectual Property Law Association (AIPLA) Report of the Economic Survey*. See https://www.aipla.org/detail/journal-issue/economic-survey-2017 for 2017.

8. Data on industry concentration (Herfindahl index and the share of the value of shipments accounted for by the 50 largest firms) were obtained from the Census Bureau. The data can be downloaded from https://www.census.gov/content/dam/Census/programs-surveys/economic-census/data/archived_tables/2007/sector31/2007_31-33_Con_Ratios_US.zip.

9. Data on patent office fees in 2011 were collected from the Office of the Federal Register, Code of Federal Regulations, Title 37: Patents, Trademarks, and Copyright. The source document can be downloaded from https://www.govinfo.gov/content/pkg/CFR-2011-title37-vol1/pdf/CFR-2011-title37-vol1.pdf or from https://www.uspto.gov/sites/default/files/aia_implementation/AC54_Final_Table_of_Patent_Fee_Changes.pdf.

10. Data on the cost of operating the patent program in 2005, 2010, and 2015 were collected from the U.S. Patent and Trademark Office (USPTO). The source documents are the USPTO's Performance and Accountability Reports, which can be downloaded from:

    2005: https://www.uspto.gov/sites/default/files/about/stratplan/ar/USPTOFY2005PAR.pdf (p65.)
    2010: https://www.uspto.gov/sites/default/files/about/stratplan/ar/USPTOFY2010PAR.pdf (p57.)
    2015: https://www.uspto.gov/sites/default/files/documents/USPTOFY15PAR.pdf (p38.)

11. Statistics on patent applications between 2011-2013 were obtained from the U.S.PTO. The data can be downloaded from https://www.uspto.gov/web/offices/ac/ido/oeip/taf/us_stat.htm.

12. Data on total R&D expenditures by all performing sectors in the U.S. in 2011 are from the National Science Foundation. See https://www.nsf.gov/statistics/infbrief/nsf14307/.

C Distance Measure

This section provides details on how we construct our patent distance metric. We describe our preferred choice, the paragraph vector approach.[1] The method consists of four steps: (1) standardizing the independent claim text, (2) turning the text into a numerical vector, (3) calculating the distances between a focal patent claim on an application to all existing granted patent claims and (4) calculating the distance to the closest existing independent claim.

C.1 Standardizing Text

The first step before converting text into a numerical vector is text standardization. We perform the following basic changes to the content of the text and remove words that carry no informational content:

1. Convert all words to lowercase
2. Narrow all sequences of words (e.g. Random Access Memory) to their acronyms (RAM)[2]
3. Remove all numbers in the text
4. Expand contractions, such as “I've” to “I have”
5. Remove all punctuation
6. Remove all stop words such as “the”, “a', “for' etc.
7. Convert all words to their base form. For example, “compromising” to “compromise”, “played” to “play,” etc.
8. Remove any punctuation within words, for example, hyphens

Once we standardize the text, we drop any claims with fewer than two words or illegible text.

To understand why this step improves the accuracy of text similarity (especially the bag-of-words approach), consider two patent claim texts, where one contains “Random Access Memory” and the other contains “RAM.” Without converting “Random Access Memory” to RAM, the algorithm will miss that the two texts are referring to the same concept and underestimate the similarity between the two claims.

[1]At the time of writing this paper, we used the state-of-the-art approach, but there is a fast-moving frontier. The most recent approaches use GPT-4 or BERT word embeddings integrated directly into Neural Networks. See Elliot and Hansen (2023) for details on text algorithms.

[2]To get data on acronyms and their meanings we scrape a website list of 250 common acronyms used in technology. The list of acronyms is available upon request.

C.2 Turning Patent Text into a Numerical Vector

We use the paragraph vector approach to represent the text of a patent claim as a numerical vector. The paragraph vector approach is an extension of the word vector approach, so we briefly explain one type of word vector method—the Continuous bag-of-words (CBOW) method—before outlining the paragraph vector approach.[3]

Word Vector Method

The CBOW approach involves predicting the word that is most likely to fill a gap in a sentence (the central word), given its surrounding words. For example, the CBOW approach involves "learning" that the most-likely word in the vocabulary to fill the gap in the sentence

`''Last week, I ______________ Spain.''`

is "visited". A by-product of this prediction task is a representation of each word as a vector, which is one step from generating a numerical representation of a paragraph.

Let D be a set of paragraphs (e.g., patent claims), denoted $d = d_1, \ldots, d_{|D|}$. Each paragraph d consists of the set of ordered words $(w_1^{(d)}, \ldots, w_{m_d}^{(d)})$. The full set of words across all paragraphs defines the vocabulary $\mathcal{V}$, of size $|\mathcal{V}|$. We represent every word $w \in \mathcal{V}$ as two vectors, $u_w \in \mathbb{R}^f$ and $v_w \in \mathbb{R}^f$, where all the elements of both vectors will be estimated. The former, u_w, is the word vector used when w is the central word. The latter, v_w, is the word vector used when w is a surrounding (contextual) word. Since there are $2f$ parameters for every word, the approach estimates $2f \times |\mathcal{V}|$ parameters, using approximately $\sum_d m_d$ data points (each being the vector of central words and their surrounding contextual words). Collect these parameters into the vector θ.

We estimate θ as the parameter vector that maximizes the log likelihood[4]

$$\sum_{d \in D} \sum_{i=k}^{m_d - k} \log \left[P(w_i^{(d)} | w_{i-k}^{(d)}, \ldots, w_{i-1}^{(d)}, w_{i+1}^{(d)}, \ldots, w_{i+k}^{(d)}; \theta) \right], \tag{1}$$

[3] We implement the Paragraph Vector approach using Gensim's `Doc2Vec` Python model (Řehůřek and Sojka, 2010).

[4] When using this objective, the approach and its optimization method is known as batch gradient descent. It is often slow in practice, and using stochastic gradient descent, which optimizes a different objective with an alternative method, is faster in practice. We present the batch gradient descent approach owing to its ease of expositition and similarity to likelihood methods in econometrics.

where k is the window size,

$$P\left(w_i^{(d)}|w_{i-k}^{(d)},\ldots,w_{i-1}^{(d)},w_{i+1}^{(d)},\ldots,w_{i+k}^{(d)};\theta\right)=\frac{\exp\left(u'_{w_i^{(d)}}\hat{v}_{w_i^{(d)}}\right)}{\sum_{w\in\mathcal{V}}\exp\left(u'_w\hat{v}_{w_i^{(d)}}\right)} \tag{2}$$

is the "softmax" likelihood of word $w_i^{(d)}$ and

$$\hat{v}_{w_i^{(d)}}=\frac{v_{w_{i-k}^{(d)}}+\cdots+v_{w_{i-1}^{(d)}}+v_{w_{i+1}^{(d)}}\cdots+v_{w_{i+k}^{(d)}}}{2k} \tag{3}$$

is the average of the relevant v terms for the words surrounding $w_i^{(d)}$. The softmax function on the right hand side of (2) is large when $u_{w_i^{(d)}}$ is close to $\hat{v}_{w_i^{(d)}}$, relative to how close $\hat{v}_{w_i^{(d)}}$ is to all other vectors u_w. Resultantly, having represented the word w as $(u_w^{(1)},\ldots,u_w^{(f)})$ (also as $(v_w^{(1)},\ldots,v_w^{(f)})$), we estimate $u_w^{(1)},\ldots,u_w^{(f)}$ and obtain the numerical vector representation $(\hat{u}_w^{(1)},\ldots,\hat{u}_w^{(f)})$.

Paragraph Vector Approach

The word vector approach delivers vector representations of all words $w\in\mathcal{V}$.[5] A naive approach to constructing a vector to represent a paragraph (e.g. patent claim) would be to average the word vectors for all words in the paragraph, $(w_1^{(d)},\ldots,w_{m_d}^{(d)})$. However, this approach for evaluating the similarity of paragraphs cannot account for sentences in context or the ordering of sentences. The Paragraph Vector approach (Le and Mikolov, 2014) offers a more sophisticated approach that builds on the word vector approach to overcome some of these issues. We describe the most straightforward analogue of the CBOW word vector method.

The idea is to carry out the CBOW word vector approach with the additional inclusion of a vector representation of a paragraph. For each paragraph $d=d_1,\ldots d_{|D|}$ we introduce a paragraph vector $x_d\in\mathbb{R}^g$. The paragraph vector x_d is included as a predictor in (1) for the words in paragraph d. An economist can think of this as a "paragraph fixed effect." To implement this in practice, the averaging in (3) is generalized to a function

$$\hat{v}_{w_i^{(d)}}=h(v_{w_{i-k}^{(d)}},\ldots,v_{w_{i-1}^{(d)}},v_{w_{i+1}^{(d)}}\ldots,v_{w_{i+k}^{(d)}},x_d),$$

typically involving concatenating or averaging the word vectors and paragraph vectors.

In using this approach, we improve the model's ability to comprehend of the meaning of words by allowing it to understand words in the context of a whole paragraph. The useful by-product it delivers is a vector representation of each *paragraph*, which was our original aim.

[5]This approach delivers *two* vector representations of words, and one can use either or average them for a single vector representation. More advanced methods like hierarchical softmax (Mnih and Hinton, 2008; Mikolov, Sutskever, Chen, Corrado, and Dean, 2013) only deliver one vector representation of a word.

C.3 Calculating Distances

The step above converts all patent claims, including those on applications and those granted, into a numerical vector. The next step involves taking every focal application patent claim vector and calculating its distance to every *existing* granted claim at the point of application. After representing a patent claim's text as a numerical vector, we use cosine similarity and angular distance, both of which are standard in the text matching and the NLP literature. We compute the cosine similarity (CS) between claim text vectors x and y as

$$cs(x,y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_j x_j^2 \sum_j y_j^2}}.$$

Then, we calculate the angular distance (AD) metric

$$AD(x,y) = \frac{\arccos(cs(x,y))}{\pi}.$$

and then double AD to obtain a normalized distance in the interval $[0,1]$.

In what follows, we describe the computational method to identify the prior art to which distance from a focal patent should be measured.

1. We split the full set of patent vectors into two separate datasets. One contains the patent claims sorted by their grant date and ID. The other contains the application claims sorted by their application date and ID. We sort on ID to ensure a unique sorting each time, since dates can clash.
2. In the full data we generate a date variable where, for granted patent claims the date is the day the patent was granted, and for the focal application claim, the date is the application day.
3. We sort the full data by the date variable and the claim ID. We assign the ids so that claims granted on the same day as an application is filled are ordered before the application. This means that applications are compared to all claims granted up to and including their application date.
4. We remove the applications along with their index in the full data. From this index, we subtract the number of prior applications. This gives a list informing us, for each application in the application full dataset, where to cut off the grant dataset so that we only compare the application claims to the prior granted claims.
5. We loop through the application dataset, filtering the grant claim dataset accordingly and calculating the cosine similarity and then angular distance to each prior granted claim.

C.4 Calculating the Distance to the Nearest Prior Art

With all distances computed, it is a simple step to find the closest 50 claims to each application. We experiment with different choices on which percentile of the closest 50 distances to use. We also experimented with taking an average of the five closest distances for example, and the resulting distances were similar.

D Descriptive Results

We show how patent application outcomes vary with technology center and examiner seniority. First, we regress a binary variable equal to one if the application process lasts more than one round against fixed effects for examiner seniority grade, technology center, year of application, and a small entity indicator (applying firm having fewer than 500 employees). The results in Column (1) of Table D.1 reveal substantial variation across technology centers; e.g., Computer Networks (TC-24) has a 12 percentage point higher likelihood of multi-round negotiation than the reference category, Biotechnology (TC-16). Further, the likelihood of any negotiation decreases with the seniority of the examiner, with senior (GS-14) examiners nine percentage points less likely to require negotiation relative to the most junior, holding technology center and application year fixed. Further, small entities are 12 percentage points less likely to negotiate (all else fixed).

In Column (2), we do the same analysis for the dependent variable equal to one if the examiner grants a patent. We match the findings of Frakes and Wasserman (2017) – senior examiners are more likely to grant and grant rates vary substantially across technology centers. In our model, we explain this variation by letting the distribution of intrinsic motivation vary with seniority level, by incorporating differences in the credit structure for examiners that vary across seniority and technology centers, and by allowing fighting costs to differ for applicants, with technology category-specific distributions. Our parameter estimates enable us to disentangle the effects of these factors in explaining the variation in outcomes, as we discuss in the text.

Table D.1. Regression Results

Variable	(1) Negotiation	(2) Grant
Intercept	0.7433 (0.006)	0.542 (0.005)
GS-7	-0.002 (0.004)	0.003 (0.005)
GS-9	-0.016 (0.004)	0.035 (0.004)
GS-11	-0.020 (0.004)	0.066 (0.004)
GS-12	-0.034 (0.004)	0.092 (0.004)
GS-13	-0.045 (0.004)	0.126 (0.004)
GS-14	-0.091 (0.004)	0.178 (0.004)
Chemicals (17)	0.064 (0.002)	0.067 (0.002)
Comp. Software (21)	0.105 (0.002)	0.196 (0.002)
Comp. Networks (24)	0.123 (0.002)	0.192 (0.002)
Communications (26)	0.047 (0.002)	0.198 (0.002)
Electronics (28)	-0.010 (0.001)	0.244 (0.001)
Other (36)	0.065 (0.002)	0.136 (0.002)
Mech Engineering (37)	0.042 (0.002)	0.139 (0.001)
Small Entity	-0.120 (0.001)	-0.170 (0.001)
Year FE	Yes	Yes
N	1,641,333	1,759,313

Notes: Omitted grade is GS-5 and omitted technology center is Biotechnology and Organic Fields (16). Technology center "Other" refers to Center 3600, which is "Transportation, Electronic Commerce, Construction, Agriculture, Licensing and Review." Following Frakes and Wasserman (2017), we omit GS-15 grade examiners. We report heteroskedasticity robust (HC1) standard errors in parentheses.

These results show stark differences in average grant rates and likelihood of negotiation across technology centers and examiner seniority grades. Next, we investigate the variation in *examiner-specific* decisions within and between seniority grades and technology center pairs. To do this, we calculate examiner-specific outcomes (average grant rates, number of rounds, length of examination period, probability of negotiation, etc.) within each seniority grade examiners are in at the time. We decompose the variation in these examiner averages into within and between seniority grade-technology center pairs by introducing dummies for each seniority-grade-technology-center dyad in Table D.2. The proportion of within-group variation in examiner grant rates is 80%, implying substantial variation in examiner grant rates not explained by seniority and technology centers. Our model explains this variation in examiner-specific grant rates within the technology center and seniority groups by incorporating group-specific *distributions* of examiner intrinsic motivation and costs of delay.

Table D.2. ANOVA Results

Variable	**Grade × TC Fixed Effects**
Grant rate	79.84
Duration of examination (years)	75.79
Number of rounds	80.89
No negotiation (one round)	89.53
Independent claims granted	74.93

Notes: For each variable y, and an examiner e when they are in seniority grade S and technology center T, we calculate $\bar{y}_{eST}$. Then we regress $\bar{y}_{eST}$ on a set of interactive dummies for seniority grade and technology center. We report $1 - R^2$ (as a percentage) for these regressions, thereby providing the proportion of within group variation.

E Model Extensions and Details

E.1 Extensions

For future research, we now show several ways that the model can be extended. We do not implement these extensions empirically in this paper.

E.1.1 Examiner Learning

The current model allows examiners to make errors but maintains the same errors throughout the negotiation stage. In this sense, there is no updating of the *error* by the examiner. However, the grounds for rejection *will* be recalculated at every negotiation round, as the applicant narrows the extent of padding in response to a rejection by the examiner.

There are two ways in which we could introduce examiner learning in an extended model. One approach is to allow the examiner to correct *part* of the error with probability one. For example, the assessment in round r would become

$$\hat{D}_j^r = \mathcal{D}(D_j^*, p_j^r) \cdot \varepsilon_j^r,$$

where

$$\varepsilon_j^r = \varepsilon_j^{r-1} - (\varepsilon_j^{r-1} - 1)\ell_\varepsilon$$

and $\ell_\varepsilon \in [0, 1]$ is the rate of examiner learning on distance. An alternative approach is to have the examiner *fully* learn with some probability. In that case, ε becomes 1 after the examiner discovers all the relevant prior art that they were missing.

E.1.2 Indefiniteness Rejections

We can extend the model to capture rejections of claims based on indefiniteness (112). In the baseline model, we do not distinguish them from novelty (102) and non-obviousness (103) rejections. To do this, we define claims as a triple $C_j = (v_j^*, D_j^*, I_j^*)$. The first two terms are as described in the baseline model. The new term I_j^* is the true ambiguity of the description of claim j. Now padding p increases claim indefiniteness from I_j^* to $\tilde{I}_j = \mathcal{I}(I_j^*, p)$, where the indefiniteness function $\mathcal{I}(\cdot, \cdot)$ is increasing in both arguments. The examiner now obtains an assessment $\hat{I}_j$ of the indefiniteness of claim j, given by

$$\hat{I}_j = \mathcal{I}(I_j^*, p) \cdot u_j,$$

where u_j is the examiner's indefiniteness error. The examiner has *grounds* for an indefiniteness rejection if $\hat{I}_j$ exceeds an indefiniteness threshold τ^I, that is $\hat{I}_j > \tau^I$. Again, this does not necessarily mean the examiner will reject, instead they just know that they have legitimate reasons to reject, should they want to.

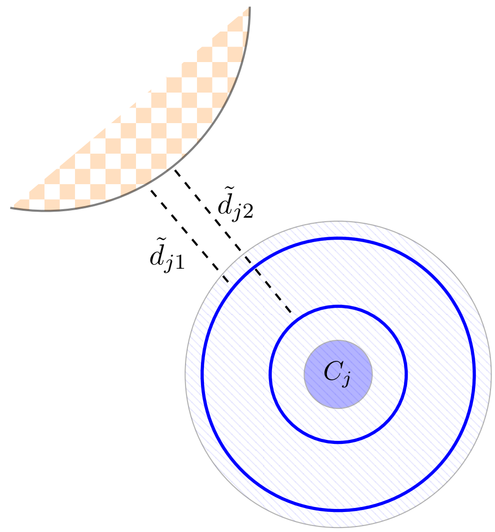
The rest of the model remains the same. The extension is more attractive in the sense that assessments of indefiniteness are claim-specific beyond the examiner error. However, whether it is worthwhile to include this extension depends on whether the data can identify a distribution of true indefiniteness I_j^*. One possible approach is to use algorithms that estimate the "readability" of patent claims text, as in Kong, Dulleck, Jaffe, Sun, and Vajjala (2023).

E.1.3 Dependent Claims

In this subsection we show how the model can be extended to include dependent claims. There are m_j^0 dependent claims associated with independent claim j. Dependent claim k is distance $\tilde{d}_{jk}$ from the nearest existing point in the intellectual property space. Independent claims mark the boundaries of the different aspects of the invention, and dependent claims identify successively narrower interpretations of the independent claim. Figure E.1 shows how to visualize dependent claims, similar to Figure 2 in the text.

Padding now changes the distances of dependent claims, as well as independent claims. Padding

FIGURE E.1. DEPENDENT CLAIMS



Notes: The Figure represents dependent claims by the thick rings inside the padded independent claim. We see two dependent claims here, with dependent claim 1 broader than dependent claim 2. The two dependent claims are distances $\tilde{d}_{j1}$ and $\tilde{d}_{j2}$ from the nearest point in the space of prior art.

determines the values of dependent claim distances $\tilde{d}_{jk}$ by splitting up the interval between unpadded distance D_j^* and padded distance $\tilde{D}_j$ (i.e. $[\tilde{D}_j, D_j^*]$) into m_j^0 blocks:

$$\tilde{d}_{jk} = \tilde{D}_j^0 + \frac{k}{m_j^0 + 1}\left(D_j^* - \tilde{D}_j^0\right) \qquad k = 1, \ldots, m_j^0.$$

If an independent claim is rejected, so are all the associated dependent claims. To reduce padding on rejected claims, the applicant cancels any dependent claims k such that $\tilde{d}_{jk} < \tilde{D}_j^{t+1}$. Finally, the applicant's payoff if the examiner grants the payoff is the same as in the baseline model, except for the addition of the term

$$\lambda \sum_{j=1}^{M^{GR}} \omega_j^{GR} m_j^{GR},$$

where M^{GR} is the number of granted independent claims, m_j^{GR} is the number of granted dependent claims associated with independent claim j, λ is the shadow price of dependent claims (in terms of their payoff function), and ω_j^{GR} are weights. Hence the value of dependent claims is given by a weighted sum of the number of dependent claims associated with independent claim j. The natural weights ω_j^{GR} would be $\frac{\tilde{v}_j}{\sum_k \tilde{v}_k}$, which represent the value of padded claim j relative to the total value of the padded claims. One way to interpret this payoff term is as a monetary equivalent of "insurance" in litigation after issuance. The court judges the validity of *claims*. Dependent claims allow the court to invalidate those parts of an independent claim deemed invalid without invalidating the entire independent claim. Interviews with legal scholars confirmed this interpretation of the value of dependent claims.

E.2 Examiner Credit Structure

Here we provide expressions for $g^r_{GR}(S,T)$, $g^r_{ABN}(S,T)$, $g^r_{RCE}(S,T)$ and $g^r_{REJ}(S,T)$. For $y \in \{GR,\ ABN,\ REJ,\ RCE\}$, we write $g^r_y(S,T) = \nu^r_y \cdot c(S,T)$, and give expressions for ν^r_y and $c_(S,T)$ separately.

E.2.1 Credits

Granting in the first round gives the examiner a payoff of $\nu^1_{GR} = 2$ credits. Rejecting in the first round gives $\nu^1_{REJ} = 1.25$. If the applicant abandons in round one, the examiner obtains $\nu^1_{ABN} = 0.75$. Granting in the second round gives $\nu^2_{GR} = 0.75$ credits. Rejecting in the second round gives $\nu^2_{REJ} = 0.25$ credits, with an extra $\nu^2_{ABN} = \nu^2_{RCE} = 0.5$ credits whether the applicant abandons or continues to an RCE. Ultimately, the examiner obtains two credits irrespective of what happens in the first two rounds. The only difference is whether they obtain the credits immediately (say, from an immediate grant) or spread out over two rounds.

The structure of the payoffs in the first RCE are the same, except $\nu^3_{REJ} = 1$ and $\nu^3_{GR} = 1.75$. In this case, irrespective of what happens in the RCE, the examiner will obtain 1.75 credits. The difference comes from whether they receive all 1.75 credits at once by granting, or 1 credit from their non-final rejection and $\nu^4_{REJ} = 0.25$ plus $\nu^4_{ABN} = \nu^4_{RCE} = 0.5$ credits from the applicant's response.

In the second and any subsequent RCEs, the structure of the payoffs is still the same, except $\nu^{2r+1}_{REJ} = 0.75$ and $\nu^{2r+1}_{GR} = 1.5$ $(r > 1)$. As before, the examiner will receive 1.5 credits from second and subsequent RCEs. The difference comes from whether they receive all 1.5 credits at once from granting, or 0.75 credits from their non-final rejection and $\nu^{2r+2}_{REJ} = 0.25$ plus $\nu^{2r+2}_{ABN} = \nu^{2r+2}_{RCE} = 0.5$ credits from the applicant's response.

E.2.2 Seniority and Technology Complexity Adjustments

The seniority and technology complexity adjustment term is

$$c(S,T) = \frac{c_{TECH}(T)}{c_{SEN}(S)}.$$

Table E.1 gives the values of $c_{SEN}(S)$ across the GS categories. Higher seniority factors imply larger values of c_{SEN}, and therefore lower values of credits. Table E.2 gives the values of $c_{TECH}(T)$

we created for the different technology centers and use in the estimation of the model. The Patent Office does not have adjustments at the technology center level, but rather at the more detailed U.S. Patent Class (USPC) level. We obtained the adjustments at the USPC level from the Patent Office and constructed a patent-application weighted average for each technology center.

Table E.1. Seniority Corrections

Seniority Grade	**Signatory Authority**	$c_{SEN}(S)$
GS-5	None	0.55
GS-7	None	0.7
GS-9	None	0.8
GS-11	None	0.9
GS-12	None	1.0
GS-13	None	1.15
GS-13	Partial	1.25
GS-14	Partial	1.25
GS-14	Full (primary examiner)	1.35

Notes: This table provides the seniority factors for credit adjustment. In the empirical work, we use 1.15 for GS-13 and 1.25 for GS-14.

Table E.2. Technology Center Adjustments

Technology Center T	**U.S.PTO Number**	**Correction** ($c_{TECH}(T)$)
Chemical and Materials Engineering	17	22.2
Computer Architecture Software and Information Security	21	31
Computer Networks, Multiplex, Cable and Cryptography/Security	24	29
Communications	26	26.5
Semiconductors, Electrical and Optical Systems and Components	28	21.4
Transportation, Electronic Commerce, Construction, Agriculture...	36	22.4
Mechanical Engineering, Manufacturing and Products	37	19.9

F Computational Details

F.1 Logic of Structural Estimation Code

The basic logic of the code for estimating the structural model is as follows:

1. Draw values for the fundamental shocks (e.g., application fighting cost *percentiles* and realized examiner types), which we use for *all* parameter guesses.

2. **Outer loop optimization (Julia `BlackBoxOptim`)**:
   PURPOSE: SOLVING FOR THE PARAMETER VECTOR
   Fix a value of the parameter vector ψ:

   (a) **Inner loop:**
   PURPOSE: CALCULATING THE MODEL MOMENTS FOR A GIVEN VALUE OF ψ
   For each simulated draw $s = 1, \ldots, \mathcal{S}$:

      i. **Optimal initial padding:**
      PURPOSE: CALCULATING EXPECTED PAYOFFS FOR EVERY CHOICE OF p_0
      Use the `GoldenSection` univariate optimization method to solve for the applicant's optimal initial level of padding

      ii. With that choice of initial padding:

         A. Work out if the applicant will choose to apply

         B. If the applicant does apply, solve the game for the *realized* examiner type and errors for which the prosecution process is actually conducted under

      iii. If the applicant does apply, store the vector of endogenous variables for this simulation, $\mathbf{m}_s(\psi)$

   (b) Calculate the simulated moments $\bar{\mathbf{m}}(\psi)$ from $\mathbf{m}_1(\psi), \ldots \mathbf{m}_{\mathcal{S}}(\psi)$

   (c) Compute the value of the objective $(\bar{\mathbf{m}}(\psi) - \mathbf{m}_{\mathcal{S}})'\Omega(\bar{\mathbf{m}}(\psi) - \mathbf{m}_{\mathcal{S}})$ where $\mathbf{m}_{\mathcal{S}}$ is the vector of empirical moments from the data

F.2 Optimization Details

We use Julia's `BlackBoxOptim` library to estimate the model parameters. The optimization routine has four main options, which we choose as follows:

1. *Search Range*: Table F.1 provides the parameter bounds we search over.

2. *Maximum Function Evaluations:* We use a limit of 120,000 evaluations of the objective function in our search. We ran the baseline model with up to 400,000 evaluations and the estimates and objective function were quantitatively unchanged.

3. *Method*: We use the default "adaptive_de_rand_1_bin_readiuslimited" method. This is an Adaptive Differential Evolution optimizer with radius limited sampling.

4. *Population Size*: We set the initial population size of particles for the routine to 100.

We optimize over initial padding choices using Julia's `Optim` library. We use the `Golden Section` method, which is a special method for univariate optimization without derivatives. We set an absolute tolerance of 0.02.

TABLE F.1. PARAMETER BOUNDS

Parameter	**Symbol**	**Lower**	**Upper**
Junior intrinsic motivation log-mean	$\mu_{\theta,\text{junior}}$	0.05	5
Senior intrinsic motivation log-mean	$\mu_{\theta,\text{senior}}$	0.05	5
Intrinsic motivation log-sigma	σ_θ	0.1	2
Examiner delay cost log-mean	μ_π	0.1	4
Examiner delay cost log-sigma	σ_π	0.1	2
Error, standard deviation	σ_ε	0.01	0.3
Initial returns log-mean	μ_v	5	11
Initial returns log-sigma	σ_v	0.5	5
Initial distance, alpha	α_D	1	8
Initial distance, beta	β_D	1	8
Narrowing probability	η	0	1
Application obsolescence probability	$P_{w,\text{app}}$	0	1
Renewal obsolescence probability	$P_{w,\text{renew}}$	0	1
Fighting cost log-mean	μ_f	5	10
Fighting cost log-sigma	σ_f	0.1	5

Notes: Bounds for fighting costs are the same for simple and complex applications (as provided in the final two rows).

G Moment Selection

First, we provide further details on the possible moments we could use to estimate our model. Then, we provide some information on our methods to prune moments from the full set.

G.1 Available Moments

We have seven sets of moments available, which we describe in turn.

Our first group of moments corresponds to examiners' issuance and applicants' abandonment decisions. For each round in the model and each seniority level, we calculate the proportion of applications examiners grant and the proportion that applicants abandon. Since there are nine seniority grade-signatory authority pairs, and we observe at least six rounds, this implies at least 108 moments on grants and abandonments.

Second, we observe the distribution of the proportion of claims rejected, both by round (six) and by seniority grade-signatory authority pair (nine). These observations generate another 54 moments.

Third, we observe the proportion of granted patents that renew at four, eight, and twelve years after issuance. These observations generate four moments on patent renewals (don't renew at four, renew at four but not eight, renew at eight but not twelve and renew at twelve).

Fourth, we calculate the distribution of claim distances by round. We calculate the mean and standard deviation of the distance distribution by round for at least six rounds, implying at least 12 moments on distance. Another moment comes from the within-application distance correlation.

Fifth, at each of the nine seniority grades, we calculate each examiner's *leniency*, which is their average rejection rate across all the applications they examine. Hence for each seniority grade-signatory authority pair, we obtain a distribution of examiner rejection rates, for which we can calculate the mean and standard deviation of the distribution of examiner fixed effects. From this we obtain another 18 moments.

Next, given that we can identify the distance threshold externally, we calculate the proportion of granted patents containing at least one invalid claim (that is, a claim whose distance is below the distance threshold). Hence, for each round and each seniority level, we calculate the proportion of patents granted containing an invalid claim, implying another 54 moments.

Finally, we observe the distribution of application fighting costs. We have six moments on the distribution of legal application fees for four technology categories (simple, chemical, electrical and mechanical), which we match to the technology centers on which we estimate the model. This implies another 24 moments.

G.2 Choosing Moments

We have more than two hundred data moments that we can calculate from endogenous variables in the model. Since we have 21 model parameters to estimate with simulated method of moments, in principle, we are over-identified. However, not all moments will aid the estimation procedure in identifying the parameters, so we begin by pruning the set of moments for estimation.

We follow a rigorous, data-driven methodology to create a subset of the moments that best estimate the parameters. To do this, we calculate the sensitivity matrix described in Andrews, Gentzkow, and Shapiro (2017). As the authors explain, "sensitivity gives a formal, quantitative language in which to describe the relative importance of different moments for determining the value of specific parameters." If a moment had a small value in the sensitivity matrix for all parameters, we considered it as not useful in estimating our model. Further, as described in Jalali, Rahmandad, and Ghoddusi (2015), for each parameter and moment, we plot the value of the moment for different values of the parameter, fixing the other parameters at their estimates. If this curve is flat, this parameter does not influence on the value of the moment. For a given moment, if the curve is flat across *all* parameters, it suggests that the moment offers no useful variation to identify the parameters.

For each parameter, we also plot the value of the SMM objective across all values of the parameter, fixing other parameters at their estimates. Ideally, the SMM will be U-shaped in each parameter to ensure a well-defined global minimum exists. By doing this, we learn how well we pin down parameters based on the set of moments we have available.

By combining the sensitivity matrix with moment and SMM plots, we pruned the set of moments down to those that offer some assistance in estimating the parameters. Since we split many parameters into two seniority groups (junior and senior), we split some of our moments into the same seniority categories.

H Quantification of Social Costs

H.1 Implementing Type 1 Social Cost Calculation

As indicated in the text, a key challenge in implementing our calculation of type 1 social costs comes from the fact that the estimates of the value of patent rights for invalid patents include potential litigation costs. To impute the "value at stake" in litigation for these patents, we need to adjust our methodology to exclude these costs.

To do this, we make two assumptions:

A1: Valid patents are not litigated. This assumption holds in a model with perfect courts, where a competitor knows (or can pay a fee to discover) whether a patent is valid or not, and then choose whether to litigate based on the result.[6] This assumption allows us to calculate the value of patent rights for valid patents, $\tilde{V}$, as equal to the observed value since there are no litigation costs to net out.

A2: The *distribution* of the value at stake, $G_{\tilde{V}}(\cdot)$, is the same for valid patents as invalid patents. The basis for this assumption is that initial distances and values are uncorrelated in the model. This assumption allows us to draw values from the observed distribution of $\tilde{V} = V$ for valid patents and use them as draws from the distribution of $\tilde{V}$ for invalid patents.

Given A1 and A2, the procedure for calculating type 1 social costs is as follows:

1. Estimate the parameters of a log-normal distribution for the value at stake for *valid* patents.[7] Let the estimated distribution be denoted as $\hat{G}_{\tilde{V}}(\cdot)$.

2. Let $\bar{P}$ be the total number of *invalid* patent grants for the given period we simulate. Then, for each $p = 1, \ldots, \bar{P}$:

   (a) Take a draw from the estimated distribution of *valid* patents' value at stake (ex post value), $\hat{G}_{\tilde{V}}(\cdot)$, to represent the value at stake for the invalid patent p

   (b) Using the draw, calculate S_{1p} from Equation (14).

3. Calculate the total social cost of type 1 error as $\sum_{p=1}^{\bar{P}} S_{1p}$.

Finally, note that we calculate the threshold for exposure to litigation from the *empirical* distribution of the value at stake for valid patents, $\hat{G}_{\tilde{V}}(\cdot)$.

H.2 Implementing Type 2 Social Cost Calculation

The primary challenge in implementing our calculation of type 2 social costs comes from calibrating the value of the invention without patent rights (π), particularly for inventions with $\Gamma^* \leq 0$, where we cannot use the patent premium. In a similar vein to our approach to type 1

[6] This assumption is *not* at odds with Schankerman and Schuett (2022), where *high types* are litigated with some probability even though they will not be invalidated. The important point is that high types in their model (patents that would not be developed without patent rights) are not the same as valid patents in our model, which are defined as those with distance larger than the threshold.

[7] The sum of log-normal distributions is approximately log-normal (Dufresne, 2004), which our simulation here exhibits.

social costs, we assume that the distributions of π for those with positive and negative Γ^* are the same and then draw values of π from this distribution for those inventions.

To be precise, our specific implementation is as follows:

1. Draw a pilot set of potential inventions, used to calculate a distribution of π. Run these set of potential inventions through the model and calculate Γ^*. For those with positive Γ^*, create a distribution of π using the relationship $\Gamma = \xi\pi$.

2. Now start the simulation for type 2 social costs by drawing a new set of potential inventions (returns, distances, number of claims, fighting costs, examiner etc.). For each potential invention i, calculate Γ_i^*. If $\Gamma_i^* > 0$, calculate $\pi_i = \frac{\Gamma_i^*}{\xi}$. If $\Gamma_i^* \leq 0$, draw a value of π_i from the distribution calculated in 1. Also, draw a development cost κ_i.

3. For each of the potential inventions i, work out the set $i = 1, \ldots, \mathcal{I}_{\text{no dev}}$ that do not develop as those with $\max\{\Gamma_i^*, 0\} + \pi_i < \kappa_i$

4. For $i = 1 \ldots, \mathcal{I}_{\text{no dev}}$, run the potential invention through a model where, at the point of abandonment, the inventor obtains all valid claims they have, and so obtains the patent value of their valid claims, instead of a payoff of 0. By definition, this scenario has the property that all abandoned claims are invalid, so that there is no type 2 error. Let Γ_i' denote the expected value of patent rights in this new scenario.

5. For $i = 1 \ldots, \mathcal{I}_{\text{no dev}}$, calculate the set $i = 1, \ldots, \mathcal{I}_{\text{now dev}}$ who have $\max\{0, \Gamma_i'\} + \pi_i \geq \kappa_i$. This is the set who do not develop because of type 2 error but do develop in the absence of type 2 error.

6. For $i = 1, \ldots, \mathcal{I}_{\text{now dev}}$, calculate $S_{2i} = \frac{\rho_{\text{soc}}}{\rho_{\text{priv}}}\Big(\max\{0, \Gamma_i\} + \pi_i\Big) - \kappa_i$ and calculate the total type 2 social cost as
$$T_2 = \sum_{i=1}^{\mathcal{I}_{\text{now dev}}} S_{2i}.$$

H.3 Calibrating Deadweight Loss

In the derivation of deadweight loss, note that
$$DWL = \frac{1}{2}\Delta\wp\Delta q = \frac{1}{2}\frac{\Delta q}{q}q\Delta\wp = \frac{\lambda}{2}\frac{\Delta\wp}{\wp}\tilde{V},$$
by the definitions of $\tilde{V}$ and λ. Further, note that
$$\frac{\Delta\wp}{\wp} = \frac{q\Delta\wp}{q\wp} = \frac{\text{lic. rev}}{\text{sales}} = \frac{\text{lic. rev}}{\text{R\&D}} \cdot \frac{\text{R\&D}}{\text{sales}}$$

As described in the text, we use Schankerman and Schuett (2022) for the ratio of licensing revenue to R&D, and data from the Bureau of Economic Analysis for the ratio of R&D to sales.

H.4 Deadweight Loss Under Cournot Competition

In the main text, we compute deadweight loss from a patented invention assuming symmetric licensees operate in a perfectly competitive industry. Suppose instead that the licensees compete in a Cournot setting. By standard calculations, the equilibrium price-cost margin is $\dfrac{\wp - c}{\wp} = \dfrac{m^*}{\lambda}$ where $m^* = \dfrac{1}{N}$ is the average market share and λ is the demand elasticity. We write this as $\dfrac{\wp - c}{\wp} = \frac{H^e}{\eta}$ where H^e is the symmetric-equivalent Herfindahl index of concentration. Thus for $H^e < 1$

$$\wp = \frac{c}{1 - \frac{H^e}{\lambda}}.$$

With imperfect competition, the change in equilibrium price is larger than the Arrow royalty due to double marginalization: $\Delta\wp = \frac{\Delta c}{1 - \frac{H^e}{\lambda}} > \Delta c$. The associated deadweight loss with Cournot competition is

$$\begin{aligned}
DWL_{\text{cournot}} &= \frac{1}{2}\Delta\wp\Delta q \\
&= \frac{1}{2}\frac{\Delta c}{1 - \frac{H^e}{\lambda}}\Delta q \\
&= \frac{1}{2}\frac{1}{1 - \frac{H^e}{\lambda}}\frac{\Delta q}{q} \cdot q\Delta c \\
&= \frac{1}{2}\frac{1}{1 - \frac{H^e}{\lambda}}\lambda\frac{\Delta\wp}{\wp} \cdot \tilde{V}, \\
&= DWL_{\text{pc}} \cdot \frac{1}{1 - \frac{H^e}{\lambda}},
\end{aligned}$$

where it should be noted that in this case $\tilde{V} = q\Delta c$ denotes total *royalty payments*. Since $H^e \in (0, 1)$ and we require that $|\lambda| > 1$, deadweight loss in this imperfect competition setting is larger than in perfect competition case.

Using U.S. Census data for 2007, the value added weighted-average Herfindahl index for manufacturing industries (based on the 50 largest firms), H, for manufacturing sectors is 0.05. As is well-known,the Herfindahl index can be decomposed as $H = \frac{1}{N} + N \cdot \text{Var}(m) = H^e + N \cdot \text{Var}(m)$, where m is the market share of each firm. Thus, the observed H overstates the unobserved H^e, so the computed deadweight loss will be an upper bound to the true value of DWL. Despite this, the upper bound for the Cournot setting is not materially different from the competitive case in the text.

The value of H varies widely across industries. We do not compute deadweight loss using industry-specific values because it is difficult to assign patents in different patent classes to industries, and the existing Patent Office concordance is problematic (e.g., the mapping is not unique).

H.5 Calibrating Litigation Costs

To calibrate litigation costs, $\mathcal{C}(\tilde{V})$, we use data from the American Intellectual Property Law Association (AIPLA) surveys on litigation costs as a function of (intervals) of the value at stake, which we assume is the same for the patentee and challenger. We use the linear specification

$$\mathcal{C}(\tilde{V}) = \ell_0 + \ell_1 \tilde{V}$$

Using this same specification, Schankerman and Schuett (2022) estimate $\ell_0 = \$624,000$ and $\ell_1 = 0.162$ (2018 USD). Note that this calibration of legal costs is at the patent, not claim, level.

H.6 Calibrating Development Costs

We apply the estimates from Schankerman and Schuett (2022) to our context. They assume that development costs κ are exponential, with mean equal to $k_0 + k_1 s$, where s is the size reduction of the invention and k_0 and k_1 are estimated as 254.6×10^3 and 2.33×10^{10}, respectively. Regarding the size reduction, they assume that s is log-logistic distributed with parameters $\beta_0 = 1.02$ and $\beta_1 = 1.14 \times 10^{-6}$. We use the mean value of s in our calibration.

In the baseline quantification, we draw values of κ from the distribution described above, which assumes that development costs are independent of Γ^* and π. In this model, inventors know their development costs prior to their decision to develop their idea. We also experiment with another model, which makes the opposite assumption that inventors do not know their development costs and thus use the mean value, $\bar{\kappa} = k_0 + k_1 \bar{s}$, to make their development decision. Both models produce similar conclusions; results are available upon request.

H.7 Calibrating the Number of Ideas

To compute the number of ideas, we start with the average annual number of utility patent applications in the period 2011–2013. We convert this number into the number of ideas in two steps. First, we use the estimates from Schankerman and Schuett (2022) that about two-thirds of applications are “low type” inventions (defined by them as those that would have been developed even without patent protection), and second, that one-third of ideas become a low type patent application. Together, this implies about one million ideas for potential inventions for each cohort of applications.

References

Andrews, I., M. Gentzkow, and J. M. Shapiro (2017): "Measuring the Sensitivity of Parameter Estimates to Estimation Moments," *The Quarterly Journal of Economics*, 132, 1553–1592.

Dufresne, D. (2004): "The Log-Normal Approximation in Financial and Other Computations," *Advances in Applied Probability*, 36, 747–773.

Elliot, A. and S. Hansen (2023): "Text Algorithms in Economics," *Unpublished Working Paper*.

Frakes, M. D. and M. F. Wasserman (2017): "Is the Time Allocated to Review Patent Applications Inducing Examiners to Grant Invalid Patents? Evidence from Microlevel Application Data," *The Review of Economics and Statistics*, 99, 550–563.

Jalali, M., H. Rahmandad, and H. Ghoddusi (2015): *Using the method of simulated moments for system identification*, MIT Press.

Kong, N., U. Dulleck, A. Jaffe, S. Sun, and S. Vajjala (2023): "Linguistic Metrics for Patent Disclosure: Evidence from University versus Corporate Patents," *Research Policy*, 52.

Le, Q. and T. Mikolov (2014): "Distributed representations of sentences and documents," in *International conference on machine learning*, PMLR, 1188–1196.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013): "Distributed Representations of Words and Phrases and their Compositionality," *Advances in Neural Information Processing Systems*, 26.

Mnih, A. and G. Hinton (2008): "A Scalable Hierarchical Distributed Language Model." *Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference*, 1081–1088.

Řehůřek, R. and P. Sojka (2010): "Software Framework for Topic Modelling with Large Corpora," *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50.

Schankerman, M. and F. Schuett (2022): "Patent Screening, Innovation, and Welfare," *The Review of Economic Studies*, 89, 2101–2148.