Online Appendix for: "*Enforcement Policy in a Dynamic Theory of Deterrence*"

Clifford Bekar* Kenneth I. Carlaw† B. Curtis Eaton‡

Revised September 16, 2023

*Associate Professor, Economics, Lewis and Clark College, bekar@lclark.edu.

†Professor, Economics, University of British Columbia - Okanagan, kenneth.carlaw@ubc.ca.

‡Emeritus Professor, Economics, University of Calgary and Honorary Professor, Economics, University of British Columbia - Okanagan, eaton@ucalgary.ca.

Contents

1 Replication Materials

This repository consists of material and code for replicating the quantitative results and Figures from *"Enforcement Policy in a Dynamic Theory of Deterrence"*. Also included are a range of robustness tests referenced in the text.

We have implemented most of our numerical simulations in two wholly separate codebases, *Matlab* and *Python.* Our reported results have been independently replicated in each codebase. One member of our team wrote and maintained the Matlab codebase, another wrote and maintained the Python codebase.

All code can be found at the following Github repository:

- Python file(s): Python Codebase
- Matlab file(s): Matlab Codebase

1.1 Notes on the simulator

We now describe the simulator and our benchmarking efforts.

Initiating a simulation requires values for: the quantity of enforcement resources (R); people's look back horizon (z); the number of potential violators (N); the mean (μ) and standard deviation (σ) of $\phi(g)$ the PDF from which each persons private gains from committing a violation is drawn; the probability a violator is apprehended if investigated (γ); the utility penalty paid by apprehended violators (F); and, the population's Bayesian priors regarding apprehension (α, β). Unless noted, we set these parameters according to our baseline parameterization in both codebases.

The number of states in $\boldsymbol{Z}$ for $z = 1$ and $N \leq 50$ is manageable, meaning it is feasible to produce the associated transition matrix $\boldsymbol{T}$ and use it to directly compute $\boldsymbol{D}$. But for $z > 1$ and larger N we must use numerical simulation methods to estimate $\boldsymbol{D}$.

We benchmark the simulator by ensuring it replicates $\boldsymbol{T}$ for a manageable set of parameters.

For each step of the simulation:

- Each potential violator is assigned a $g_i^t, \forall i \in N$ drawn randomly from $\phi(g)$ with $\mu = 0.6$ and $\sigma = 0.2$.
- A common subjective probability of apprehension, q, is calculated using the history of violations and apprehensions, $q_i^t = \frac{\alpha + a^{t-z} + \ldots + a^{t-1}}{\beta + v^{t-z} + \ldots + v^{t-1}}$, where $\alpha = 1$, $\beta = 0.25$ and for $z = 1$, $q_i^t = \frac{\alpha + a^{t-1}}{\beta + v^{t-1}}$.
- People choose to violate if and only if their $g_i^t \geq q^t \cdot F$, where $F = 1$.
- Violators are apprehended with probability $P(R^t, v^t) = p^t = \gamma \cdot \min(1, \frac{R^t}{v^t})$, where v^t is the number of violators in the current period and $\gamma = 0.8$.
- The number of violations, apprehensions, and the sum of the gains from violating (i.e., the sum of the realized draws on $\phi(g)$ for those who violated) are recorded.

- The frequency distribution of violations and apprehensions is calculated for the current period and recorded.

- The history is updated and the simulation moves to the next period.

We calculate the frequency distribution of violations and apprehensions. The frequency distribution of violations is $\boldsymbol{FV}(nv_0^t, nv_1^t, \ldots, nv_N^t)$, where nv_j^t is the number of instances in the first t periods of the simulation in which there were exactly j violations. The frequency distribution of apprehensions is $\boldsymbol{FA}(na_0^t, na_1^t, \ldots, na_N^t)$, where na_j^t is the number of instances in the first t periods of the simulation in which there were exactly j apprehensions.

The simulation proceeds in blocks of $50,000$ ticks. At the end of each block n we use $\boldsymbol{FV}$ to calculate the relative frequency distribution of violations $\boldsymbol{RFV}^n = (rv_0^n, \ldots, rv_N^n)$, where rv_j^n is the proportion of periods in which there were exactly j violations in the first $50000 \cdot n$ periods of the simulation. As n approaches infinity $\boldsymbol{RFV}^n$ approaches $\boldsymbol{D}$. The question is: How large must n be to ensure that $\boldsymbol{RFV}^n$ is a *good* estimate of $\boldsymbol{D}$? To answer this we need a convergence criterion.

At the end of each simulation block, for all $n \geq 2$, we calculate the following test statistic:

$$TEST^n = \sum_{j=0}^{j=N} |rv_j^n - rv_j^{n-1}|$$

$TEST^n$ is a measure, decreasing in n, of the *distance* between $\boldsymbol{RFV}^n$ and $\boldsymbol{RFV}^{n-1}$. $TEST^n$ does not decline monotonically, which complicates any test of convergence. After extensive experimentation we settled on the following criterion: we require that $TEST^n \leq 0.01$ for 5 consecutive blocks of n. Denoting the minimum n for which the convergence criterion is satisfied by n^* yields our approximation of the stationary distribution, $\boldsymbol{RFV}^{n^*} \sim \boldsymbol{D}$.[1]

Our convergence simulator has some limitations over certain ranges of some key parameter values, in particular those that comprise the composite parameter $\frac{\sigma}{(\gamma \cdot F)}$ defined and analyzed in Section 2.2 of this appendix. If $\frac{\sigma}{(\gamma \cdot F)}$ is small the n required to achieve a good approximation of the stationary distribution is actually much larger than the n^* generated from the convergence criteria. This is because of the persistence of the q-attractors. For these parameterizations the simulator is highly sensitive to initial conditions, staying for a very long time (sometimes the entire simulation) in only one of the q-attractors and rarely (if ever for some convergence simulations around the cliff) transitioning to the other q-attractor, even though the stationary distribution is bi-model in these parameter values. This yields a poor estimate of the stationary distribution even though the convergence simulator may have taken a very long time to converge. More accurate estimations of the stationary distribution for such parameter values would take orders of magnitude longer, longer than is feasible to make simulation useful. It is important to note that this is not a limitation of the model, but rather of our computational simulation capabilities.

[1] An important confounding issue for finding n^* is that the *shape* of $\boldsymbol{D}$ changes dramatically as R changes. For low values of R, $\boldsymbol{D}$ is unimodal and right skewed, for high values unimodal and left skewed, for intermediate values it becomes bimodal. This renders n^* non-monotonic in R (see the last row of Table A.1).

1.1.1 Code for benchmarking the simulator

Code for benchmarking of the simulator can be found at:

- Python file(s): Simulator.ipynb
- Matlab file(s): ASB BenchSD, ASB BenchTMSD,

We employ Monte Carlo simulations and the relationship between $\boldsymbol{D}$ and $\boldsymbol{RFV}^{n^*}$ to: (i) cross validate the output from our Python and Matlab codebases; and, (ii) benchmark the accuracy of the simulator against a known transition matrix.[2] The computational intensity of producing both $\boldsymbol{D}$ and $\boldsymbol{RFV}^{n^*}$ increases geometrically with the size of the state space. It is only practical, therefore, to implement both approaches for a relatively small number of agents ($N = 50$) and a single period ($z = 1$) in the history of violations and apprehensions. For these parameters, and for each element of $R = \{5, 21, 45\}$, we generate 1000 instances of $\boldsymbol{RFV}^{n^*}$. We then construct the following distance metric:

$$ATEST = \boldsymbol{D} - \boldsymbol{RFV}^{n^*} = \sum_{j=0}^{j=N} |d_j - rv_j^{n^*}|$$

In Table 1 we report the mean, standard deviation, and maximal value of $ATEST$ along with the mean n^*. It is apparent that the two platforms deliver results that are essentially the same, and that $\boldsymbol{RFV}^{n^*}$ very closely approximates $\boldsymbol{D}$ as $n \to 300000$.

Table 1: Benchmarking the simulator

	Matlab Codebase			Python Codebase		
	Value of R					
	5	**21**	**45**	**5**	**21**	**45**
Mean of ATEST	0.0017	0.0098	0.0078	0.0018	0.0115	0.0087
Std Dev of ATEST	0.0009	0.0017	0.0011	0.0009	0.0019	0.0011
Max value ATEST	0.0061	0.0193	0.0116	0.0057	0.0198	0.0128
Mean n^*	6.0010	7.8430	7.0870	6.0010	7.8120	6.0010

1.2 Replicating Figures and Core Results

The code for replicating all figures in the paper can be found at:

1.2.1 Figures 4.1, 4.3, 4.4

Figures 4.1, 4.3, and 4.4 plot a random sample of 5000 periods of violations for each of the three models.

Code for producing Figures 4.1, 4.3, and 4.4 from Section 4 can be found here:

[2] We are, of course, implicitly testing the validity of our convergence criterion.

- Python file(s): Figures 4.1, 4.3, 4.4
- Matlab file(s): Figure 4.1 Figures 4.3, 4.4

1.2.2 Figure 4.2

Figure 4.2 plots the number, and type, of equilibria (when we assume $p^t = q^t \ \forall \ t$) as a function of R using Proposition 3.

Code for producing Figure 4.2 from Section 4 can be found here:

- Python file(s): Figure 4.2
- Matlab file(s): Figure 4.2

1.2.3 Figure 4.5

Figure 4.5 plots the expected number of violations for each of the three models as a function of R.

Code for producing Figure 4.5 from Section 4 can be found here:

- Python file(s): Figure 4.5
- Matlab file(s): Figure 4.5

1.2.4 Figure 4.6

Figure 4.6 plots a number of indices relevant to the dynamics of ASB for each of the three models as a function of R.

Code for producing Figure 4.6 from Section 4 can be found here:

- Python file(s): Figure 4.6
- Matlab file(s): Figure 4.6

1.2.5 Figure 4.7

Figure 4.7 plots $E(v^{t+1}|v^t)$ for $R = 39$, revealing the two q-attractors and the resulting BE.

Code for producing Figure 4.7 from Section 4 can be found here:

- Python file(s): Figure 4.7
- Matlab file(s): Figure 4.7

1.2.6 Figure 4.8

Figure 4.8 plots $E(v^{t+1}|v^t)$ for $R = 44$, revealing the two q-attractors and the resulting BE.

Code for producing Figure 4.8 from Section 4 can be found here:

- Python file(s): Figure 4.8
- Matlab file(s): Figure 4.8

1.2.7 Figure 4.9

Figure 4.9 plots measures of persistence for the q-attractors for selected values of R.

Code for producing Figure 4.9 from Section 4 can be found here:

- Python file(s):
- Matlab file(s): Figure 4.9

1.2.8 Figure 5.1

Figure 5.1 plots the total expected cost of managing ASB for all values of R.

Code for producing Figure 5.1 from Section 5 can be found here:

- Python file(s): Figure 5.1
- Matlab file(s): Figure 5.1

1.2.9 Figure 6.1

Figure 6.1 plots various ASB metrics illustrating the mechanics of dynamic policy for an arbitrarily chosen policy.

Code for producing Figure 6.1 from Section 6 can be found here:

- Python file(s): Figure 6.1
- Matlab file(s): Figure 6.1

1.2.10 Figures 6.2 and 6.3

Figures 6.2 and 6.3 plot $E(v^{t+1}|v^t)$ for optimal dynamic policies. Panel 1 of Figure 6.2 is the plot for the optimal dynamic policy, $(BE, R1, R2) = (35, 30, 57)$, conditioned on the recent history of v^{t-1}. Panel 2 is the plot of the corresponding static policy, $R = 30$. Panel 1 of Figure 6.3 is the plot of the optimal policy $(BE, R1, R2) = (0.745, 23, 48)$ conditioned on the recent history of the apprehension rate AR^{t-1}. Panel 2 is the plot of the corresponding static policy $R = 23$.

Code for producing Figures 6.2 and 6.3 from Section 6 can be found here:

- Python file(s): Figure 6.2
- Matlab file(s): Figure 6.2 Figure 6.3

1.3 Search Algorithm for Dynamic Policies

We use a two stage procedure to identify the optimal dynamic and refined dynamic policies. In the first stage, we use a directed search routine with a loose convergence criterion to identify the neighborhood in which the optimal policy lies, and in the second stage we use Monte Carlo methods to generate tight estimates of the cost of policies in the neighborhood.

The algorithm at the core of the directed search routine is this. For a given 2 bin dynamic policy at stage s, $CD^s = (BE^s, R_1^s, R_2^s)$, a modified policy $CD^{s+1} = (BE^{s+1}, R_1^{s+1}, R_2^{s+1})$, is generated using a three step procedure: in the first step, BE^{s+1} is chosen from the set $BE^s - 5, BE^s - 4, \ldots, BE^s + 5\}$ to minimize the cost of ASB for policy (BE^{s+1}, R_1^s, R_2^s); in the second step, R_{GB}^{s+1} is chosen from the set $\{R_1^s - 5, R_1^s - 4, \ldots, R_{GB}^s + 5\}$ to minimize the cost of ASB for policy $(BE^{s+1}, R_1^{s+1}, R_2^s)$; in the third step, R_2^{s+1} is chosen from the set $\{R_2^s - 5, R_2^s - 4, \ldots, R_2^s + 5\}$ to minimize the cost of ASB for policy $(BE^{s+1}, R_1^{s+1}, R_2^{s+1})$. For each of the 33 policies considered, we ran a convergence simulation and used it to estimate cost. The search is terminated when the criteria $(E(C|CD^s) - E(C|CD^{s+1}))/E(C|CD^s) < .02$ is met. Each of the 1000 simulations was seeded with an initial policy randomly chosen from the set $\{(BE, R_1, R_2)|0 \leq BE \leq 100, 0 \leq R_1 \leq 100, R_2 \leq R_2 \leq 100\}$ and for each we used the directed search algorithm to a generate a terminal policy. From initial policy to terminal policy, the average number of policies evaluated was 165. To identify the neighborhood of the optimal crackdown policy, we ranked the 1000 terminal policies from lowest to highest cost. The first 78 policies in this ranking, and 98 of the first 100, were in the set $\{(BE, R_1, R_2)|30 \leq BE \leq 39, 29 \leq R_1 \leq 32, 53 \leq R_2 \leq 65\}$. This is neighborhood of the optimal policy.

The search for the optimal 3 bin (or refined) dynamic policies was analogous to that for the 2 bin dynamic policy, with the differences being the need for two additional steps in the procedure to search over the two additional parameters, BE_2 and R_3, of the 3 bin dynamic policy. The algorithm for the 3 bin dynamic policy search is, given a 3 bin dynamic policy at stage s, $CD^s = (BE_1^s, BE_2^s, R_1^s, R_2^s, R_3^s)$, a modified policy $CD^{s+1} = (BE_1^{s+1}, BE_2^{s+1}, R_1^{s+1}, R_2^{s+1}, R_3^{s+1})$, is generated using a five step procedure analogous to that of the three step procedure of the dynamic policy search. For each of the 55 policies considered, we again ran a convergence simulation and used it to estimate the cost. The termination criteria for the 3 bin dynamic policy search was the same as that used for the dynamic policy search. The 1000 simulations were randomly seeded with initial refined crackdown policies chosen from the set $\{(BE_1, BE_2, R_1, R_2, R_3)|0 \leq BE_1 \leq 100, BE_1 \leq BE_2 \leq 100, 0 \leq R_1 \leq 100, R_1 \leq R_2 \leq R_3, R_2 \leq R_3 \leq 100\}$. From initial policy to terminal policy, the average number of policies evaluated was 440 for each of the 1000 simulations. The neighborhood of the optimal refined crackdown policies was determined in the same way as for the crackdown policy.

The estimate of cost that comes out of one convergence simulation is, obviously, a random variable. In the neighborhood of the optimal policy, the cost function is so flat that in order to reliably identify the optimum policy, many independent cost estimates are needed. Accordingly, for every policy in the neighborhood of the optimum we ran 150 convergence simulations and calculated the cost of ASB in the steady state for each of them. Our cost estimate is the mean value of these 150 cost estimates. Typically the standard deviation of cost for the 150 simulations is close to 0.20, and hence the standard error of the estimate is close to $0.016 = .2/\sqrt{150}$.

The code for the search algorithms and the Monte Carlo simulations over the set of policies in the neighborhood of the optimum can be found here:

- Matlab file(s): 2 Bin dynamic policy v^{t-1}
- 2 Bin dynamic policy AR^t
- 3 Bin dynamic policy v^{t-1}
- 3 Bin dynamic policy AR^t
- 2 Bin dynamic policy Monte Carlo v^{t-1}
- 3 Bin dynamic policy Monte Carlo v^{t-1}

Table 2: Estimated Costs of 2 bin Dynamic Policies in the Neighborhood of the Optimum.

		BE									
		30	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**
$\mathbf{R_1}$	**29**	131.28 **54**	130.91 **56**	130.66 **56**	130.57 **58**	130.62 **57**	130.81 **58**	131.17 **60**	131.62 **60**	132.16 **62**	132.77 **62**
	30	131.52 **55**	131.06 **55**	130.68 **56**	130.45 **57**	130.33 **56**	130.33 **57**	130.45 **59**	130.66 **60**	131.03 **61**	131.43 **62**
	31	131.97 **53**	131.11 **54**	131.11 **55**	130.76 **56**	130.52 **57**	130.37 **57**	130.35 **58**	130.41 **58**	130.58 **59**	130.82 **60**
	32	132.61 **53**	132.16 **53**	131.71 **55**	131.40 **55**	131.10 **56**	130.84 **57**	130.72 **58**	130.65 **59**	130.67 **59**	130.80 **60**

Note: The **bold** number in each cell is the associated R_2 of each policy. The standard errors for each cost estimate are in the range 0.013 to 0.017. So the deviation around each point estimate of costs in each cell is in the second decimal point (e.g. costs in the first cell range at most between 131.297 and 131.263.)

Table 2 reports the *minimum cost* of all 2 bin dynamic policies conditioned on v^{t-1}, in the neighbourhood of the optimal policy, $\{30 \leq BE \leq 39, 29 \leq R_1 \leq 32, 53 \leq R_2 \leq 65\}$.[3] The minimization is done with respect to R_2, whose value is reported in the associated cell. For example, the upper- left cell indicates that the minimum costs of Two-Bin policies in which $(BE, R_1) = (30, 29)$ is \$131.28, and the cost minimizing R_2 is 54. As the standard errors of these estimates are all very small (i.e., $\leq$ \$0.017), it is clear that the optimal policy is either $(35, 30, 57)$ or $(34, 30, 56)$. Every other cost estimate is more than 2 standard errors higher than \$130.33. Figure 1 demonstrates how flat the cost function is in this neighbourhood. The lowest cost reported in the table is \$130.80 and the highest is \$132.77, a difference of just 1.5%. It is even flatter in the R_2 dimension (not presented in the table, but shown in Figure 1). For example, the cost estimate for policy (39, 31, 60) is \$130.82, and the cost estimates for four related policies, (39,31,57), (39,31,58), (30,31,59) and (30,31,61), are all within one standard error of \$130.82.

The code that produces Figure 1 can be found here:

[3]We do not report it but a similar table is easily constructed for the 2 bin policy conditioned on AR^t.

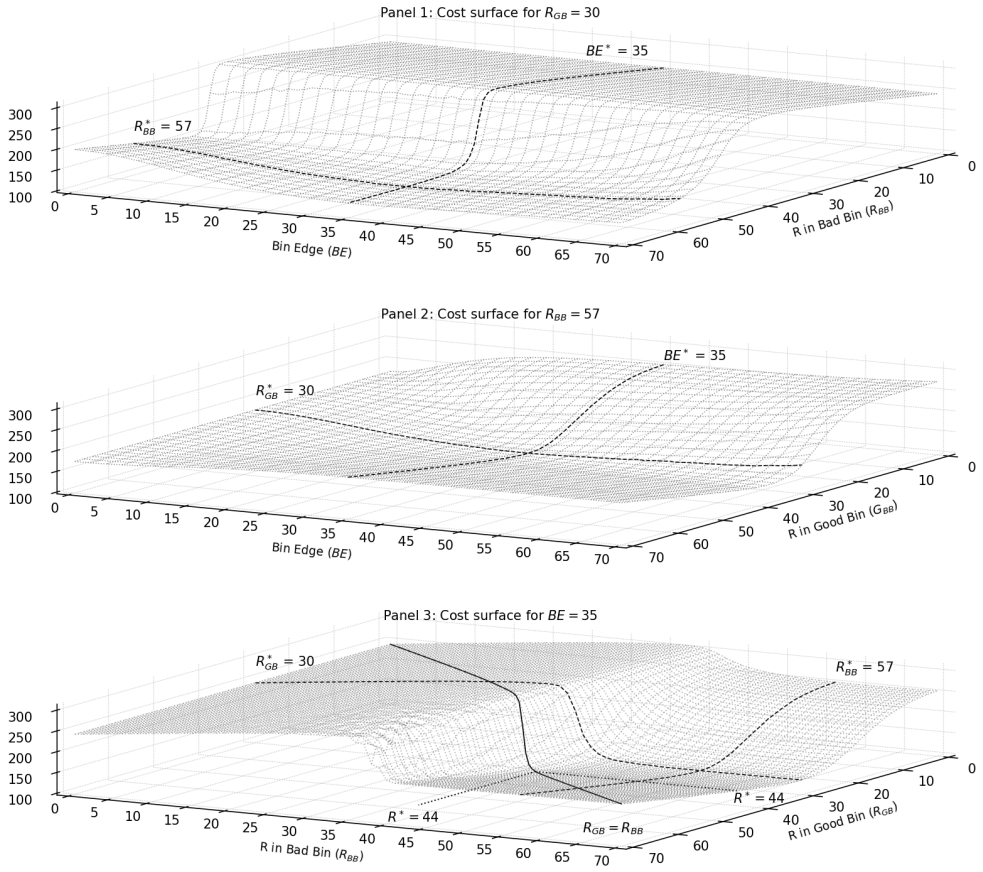

Figure 1: Surface plots of costs in BE, R_1, R_2 space

- Matlab file(s): Figure 1

Table 3 reports results of the search for the 3 bin dynamic policy search and Monte Carlo simulation for the best 25 such policies. The first two elements for a 3 bin dynamic policy, BE_1 and R_1, are the same as those for the first two elements of one of the 2 bin dynamic policies in the neighborhood of the optimum reported in Table 2. This reflects the fact that the search algorithm finds the same $BE = BE_1$ with the same R_1 for the one of the two bin dynamic policies as the 3 bin dynamic policy. Effectively the bad bin, B_2, of the two bin policy is partitioned into two bins, B_2 and B_3, with different quantities of the enforcement resource in the 3 bin dynamic policy. The 25 least costly 3 bin dynamic policies we identify in stage 1 of our search procedure are in fact refinements of one of the least costly 2 bin policies we identified in the previous section, and presented in Table 2. Key results for the 25 refined crackdown policies with the lowest tight cost estimates are reported in Table 3. Notice that the skeletons of Tables 2 and 3 are identical.

All 25 of the best 3 bin dynamic policies appear in just 11 of the 40 cells in Table 3, and 13 of them are in cells (32, 30), (33, 30) and (34, 30). In the top line of each cell we have listed the percentage difference in cost between the best 3 bin dynamic policy and 2 bin dynamic policy for that cell (from Table 2). The second line of each cell identifies the refined crackdown policy by its rank—with rank 1 being the lowest cost estimate. Policies 1, 5, 8,

Table 3: The Value of Refinement.

		BE									
		30	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**
R_{GB}	**29**	1.07 9,18	0.73 11	0.62 6	XXX	XXX	XXX	XXX	XXX	XXX	XXX
	30	XXX	XXX	0.68 2,7,15	0.49 3,4,10	0.42 1,5,8,13,14,21,22	0.17 20	XXX	XXX	XXX	XXX
	31	XXX	XXX	XXX	0.55 17	0.42 12,24	0.20 19,25	0.24 16,23	XXX	XXX	XXX
	32	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX

13, 14, 21 and 22 are listed in cell (34, 30). All are refinements of the 2 bin dynamic policy (34, 30, 56) from Table 2, and all produce a lower cost. This is not surprising – with more instruments to control ASB we expect to see a lower cost of ASB. What may be surprising is that the reduction in cost is relatively small — that is what the entry in the first line of each cell addresses. The cost estimate for the 3 bin dynamic policy with rank 1 is only 0.41% lower than the costs estimate for optimal crackdown policy (34, 30, 56).[4]

Table 4: Top ranked refined crackdown policies.

Rank	$\mathbf{BE_1}$	$\mathbf{BE_2}$	$\mathbf{R_1}$	$\mathbf{R_2}$	$\mathbf{R_3}$	**Cost**	**Std Error**
#1	34	44	30	47	64	129.79	0.013
#2	32	37	30	42	60	129.79	0.013
#3	33	43	30	49	64	129.82	0.015

In Table 4 we report the 3 bin dynamic policies that are ranked 1, 2 and 3, our costs estimates for them, and the standard errors of the cost estimates. When we look at the policies themselves we see that the refinements make intuitive sense. The lowest cost 3 bin dynamic policy is $(BE_1, BE_2, R_1, R_2, R_3) = (34, 44, 30, 47, 64)$ and it is a refinement of 2 bin dynamic policy $(BE, R_1, R_2) = (34, 30, 56)$. When 2 bin dynamic policy $(34, 30, 56)$ is used, 56 units of enforcement resource are deployed in the bad bin, $\{v^{t-1}|34 < v^{t-1} \leq 100\}$. When 3 bin dynamic policy $(34, 30 : 44, 47, 64)$ is used the bad bin is partitioned into a two two bins, the first is $\{v^{t-1}|34 < v^{t-1} \leq 44\}$ and the second is $\{v^{t-1}|44 < v^{t-1} \leq 100\}$, and quantity of enforcement resources deployed in the first is 47, somewhat less 56, and quantity deployed in the second is 64, somewhat greater than 56. In the bad bin, the proximate objective of policy is to drive violations down and back into GA, and it makes intuitive sense that the quantity of resources needed to do that efficiently is smaller when violations are close to the lower bound of BA than when violations are close to the upper bound.

To identify the optimal 2 bin dynamic policy, we generated a tight cost estimate for every policy in the neighborhood that we identified using the stage one directed search routine. That approach is not feasible for 3 bin dynamic policies. We can use the directed search

[4] As reported in the paper the best 3 bin dynamic policy conditioned on AR^t does not show any improvement over the optimal 2 bin dynamic policy.

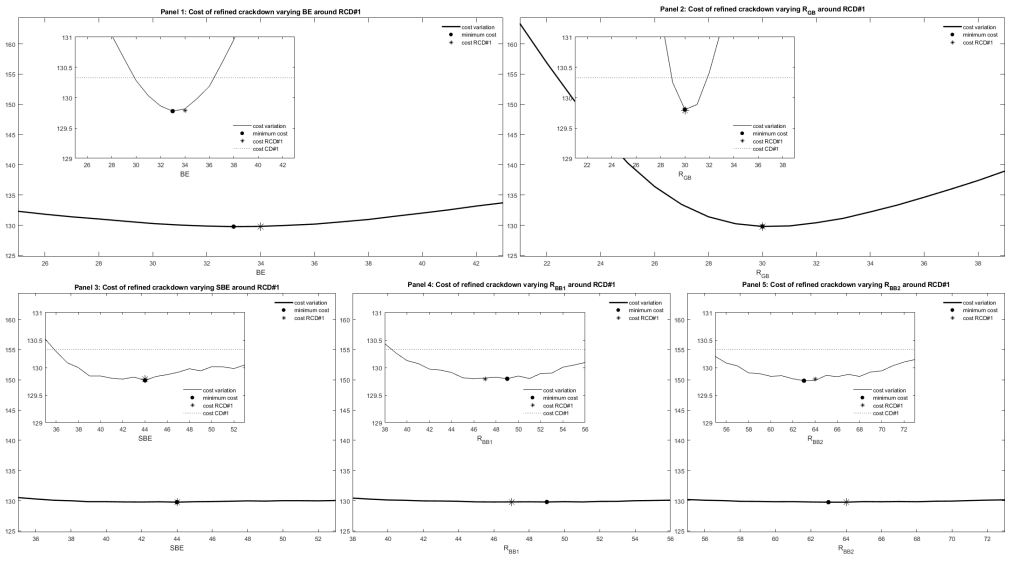

Figure 2: Cross sections of costs in the neighborhood of $RCD\#1$

routine to identify the neighborhood of the optimum, but there are so many policies in that neighborhood that it is not feasible to generate a tight cost estimate for each of them. In addition, in three of the five relevant dimensions, the cost function is so flat that it would take thousands of convergence simulations to get cost estimates that would allow us to identify the optimal policy.

In Section 6.1 of the paper, to identify the optimal crackdown policy, we generated a tight cost estimate for every policy in the neighborhood that we identified using the stage one directed search routine. That approach is not feasible for 3 bin dynamic policies. We can use the directed search routine to identify the neighborhood of the optimum, but there are so many policies in that neighborhood that it is not feasible to generate a tight cost estimate for each of them. In addition, in three of the five relevant dimensions, BE_2, R_2 and R_3 , the cost function is so flat that it would take thousands of convergence simulations to get cost estimates that would allow us to identify the optimal policy.[5]

The code that produces Figure 2 can be found here:

- Matlab file(s): Figure 2

Nevertheless, the results reported in Figure 2 suggest that the cost of ASB for the optimal policy is very close to the cost estimates in Table 4. The cost estimates reported in the figure are based on 50 independent convergence simulations for each policy examined (the standard errors of these estimates are roughly 0.03). In each panel, cost estimates are reported for 19 policies centered on policy #1: four of the five parameters of the policy are fixed at their values in policy #1, and the fifth varies up and down from its value in policy #1 by 9 units. The asterisk indicates policy #1, and the dot the policy with the lowest cost estimate. There is considerable curvature in the cost function when R_1 and BE_1 vary, which means that the

[5]This discussion is illustrated using dynamic policies conditioned on v^{t-1} but extends to dynamic policies conditioned on AR^t

optimal values of these parameters are much easier to pin down, and that is exactly what we saw in Table 4. But over the range of values examined for parameters BE_2, R_2 and R_3, the cost function is very flat and it is much more difficult to pin down the optimal values of theses parameters. This flatness reflects the fact that there are many ways to extinguish BA, and, it does matter much in terms of costs precisely how that is achieved because so little time is spent there. The inserts in each panel present the same information but the scale on the vertical axis is different. The more fine grained scale allows us to see how the cost estimates for the 3 bin dynamic policies compare to our best estimate of the minimum cost achievable with the simpler 2 bin dynamic policy—the dotted reference line in each insert represents that cost.

2 Robustness Tests

2.1 General features of the stochastic dynamic process

As we have seen, our dynamic model produces a suite of features including the cliff of positive feedback.

(i) As R increases from 0 one can distinguish three distinct regimes: initially there is an unruly regime where $E(v|R)$ is relatively high and slowly decreasing in R, followed by a transitional regime in which $E(v|R)$ drops precipitously as R increases, and finally a compliant regime where $E(v|R)$ is relatively low and virtually constant.

(ii) There is a single q-attractor in the unruly and compliant regimes. In the unruly regime the number of violations at the focal point of the q-attractor BB is relatively high and decreases slowly with R; in the compliant regime the focal point of the q-attractor GB is relatively low and unresponsive to R.

(iii) There are two q-attractors in the transitional regime: GA where the focal point violations are low and BB where the focal point violations are high. As R transits between the unruly and compliant regimes, the persistence of BA diminishes from approximately 1 to roughly 0 while the persistence of GA increases from approximately 0 to roughly 1.

(iv) Items (ii) and (iii) produce a prominent cliff-like structure in the relationship between the quantity of enforcement resources R and the expected number of violations in the stationary distribution $E(v|R)$.

(v) In all three regimes there is significant positive autocorrelation in the time series of violations for lags of one, two and three periods and for much longer lags in the transitional regime.

We develop a measure of the prominence of the cliff called $DROP$. Prominent cliffs produce large values of $DROP$, inconspicuous cliffs produce low values of $DROP$. We then define $CLIFFSET$ as the set of parameterizations such that $DROP$ exceeds a threshold value. In subsequent sections we report evidence that DROP itself is an economically interesting statistic.

2.2 CLIFFSET

The geometric signature of the cliff is a small range of contiguous values of R such that as R is increased through that range there is a dramatic drop in $E(v|R)$. We set the range of R to $RANGE = INT(0.05 \cdot N)$. For the baseline parameterization $RANGE = 5$.

For a given a parameterization we first estimate the relationship $E(v|R)$ for all $R \in \{0, ..., N\}$. For every integer $X \in \{0, ..., N - RANGE)$ we then calculate the reduction in $E(v|R)$ when R is increased from X to $X + RANGE$. $DROP$ is the maximum value of this difference, normalized by expressing it as a fraction of its maximum possible value, N.

$$DROP = \frac{\max_{\{X\}}[E(v|R = X) - E(v|R = X + RANGE)]}{N}$$

Intuitively, $DROP$ measures the height of the cliff relative to the size of the population of potential violators. It is bounded below by 0 and above by 1. For example, $DROP = 0.5$ indicates that in the vicinity of the cliff an increase of $INT(0.05 \cdot N)$ in R reduces the expected number of violations in the stationary distribution by $0.5 \cdot N$, or that the slope of the cliff is approximately -10. In defining $CLIFFSET$, we set the threshold value of $DROP$ to 0.35, producing a slope of -7. The value of $DROP$ for the baseline parameterization is roughly 0.7, producing a cliff with a slope of -14.

Figure 3 plots three different cliffs: one for the baseline parameterization, one in which $DROP = 0.35$, and in which $DROP = 0.1$. The baseline produces the most prominent cliff and the largest value of DROP.

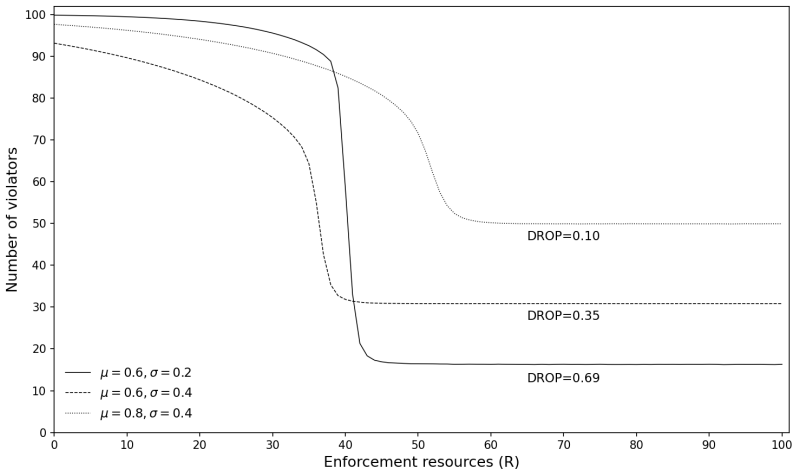

Figure 3: Three cliffs of positive feedback and their measured $DROP$.

The code that produces Figure 3 can be found here:

- Matlab file(s): Figure 3

$DROP$ is a function of six parameters, $\mu, \sigma, \gamma, F, N$ and z, so $CLIFFSET$ is an object in a six dimensional space.

$$CLIFFSET = \{\mu, \sigma, \gamma, F, N, z | DROP \geq 0.35\}$$

While a complete description is not feasible, we can describe some of the most important aspects of $CLIFFSET$. Since the underlying behavior of the model depends only on the value of $\gamma \cdot F$, and not independently on γ or F, we are able to reduce $CLIFFSET$'s dimensionality by treating $\gamma \cdot F$ as a composite parameter. Think of $\gamma \cdot F$ as the size of the deterrence stick. We start by describing a subset of $CLIFFSET$,

$$CLIFFSET' = \{\mu, \sigma \mid DROP \geq 0.35,\ \gamma \cdot F = 0.80,\ N = 100,\ z = 2\}$$

In Figure 4, we have constructed a surface plot of $DROP$ for $\mu \in \{0, 0.05, 0.1, ..., 1.00\}$ and $\sigma \in \{0.15, 0.20, ..., 0.55\}$ (so the set of (μ, σ) pairs has 189 elements).[6] We have included a shaded reference plane representing our threshold value of $DROP$. The outer boundary of $CLIFFSET'$ is defined by the locus of points where the surface plot and the reference plane intersect. We have projected the iso-DROP locus for $DROP = 0.35$ onto the baseplane, $CLIFFSET'$ is the set of (μ, σ) pairs on and inside the iso-DROP locus.

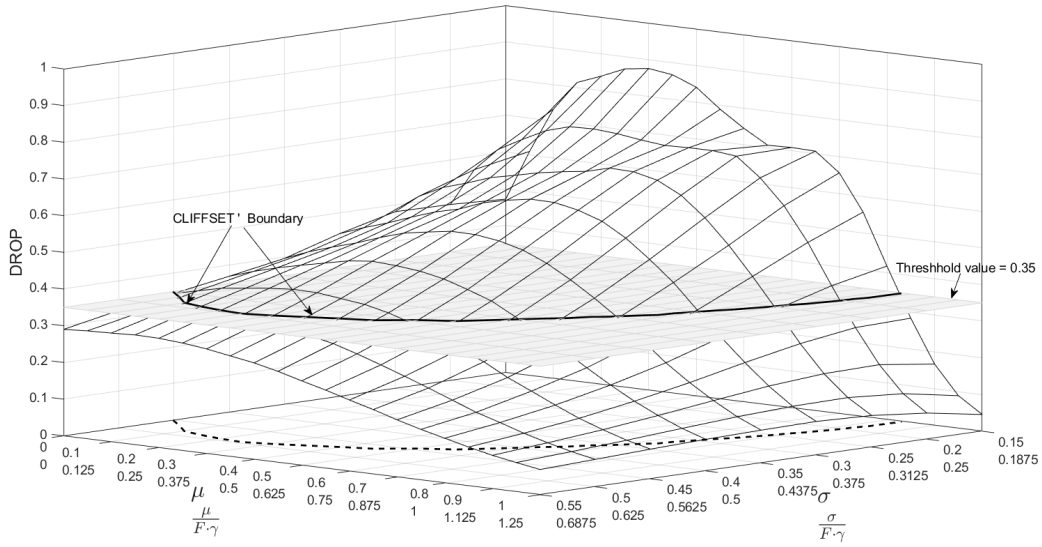

Figure 4: $DROP$ for a range of μ and σ.

The code that produces Figure 4 can be found here:

- Matlab file(s): Figure 4

There is another interpretation of $CLIFFSET'$. Setting $s > 0$ and applying this scalar to three key parameters of $CLIFFSET'$, $(s \cdot \mu, s \cdot \sigma, s \cdot \gamma \cdot F)$, the ratio of the new set of values for $DROP'$ and the original $DROP$ values are, to a very close approximation, equal to 1. In other words, $DROP$ is homogeneous of degree 0 ($HD0$) in μ, σ and $\gamma \cdot F$. This follows from the fact that the probability π that any person chooses violate in any period is $HD0$ in these same parameters. In every period, every person has a well defined subjective

[6] We have not included estimates of DROP for values of σ less than 0.10 because our convergence simulation routine is not reliable for small values of σ.

probability of apprehension, q. When their opportunity g is revealed they choose violate if and only if $g > q \cdot \gamma \cdot F$. Prior to the realization of g, the probability π that they choose violate is $1 - \Phi(q \cdot \gamma \cdot F)$. Since $\phi(g)$ is normal, $\Phi(q \cdot \gamma \cdot F)$ and hence π, are themselves $HD0$ in μ, σ and $\gamma \cdot F$.

Proposition 1 *Given any parameterization of the model* $(\widehat{\mu}, \widehat{\sigma}, \widehat{\gamma} \cdot \widehat{F}, \widehat{z}, \widehat{N}, \widehat{R})$ *and the constructed parameterization* $(s \cdot \widehat{\mu}, s \cdot \widehat{\sigma}, s \cdot \widehat{\gamma} \cdot \widehat{F}, \widehat{z}, \widehat{N}, \widehat{R})$*, where* $s > 0$*, the expected choices of potential violators are identical.*

In light of proposition 1 a simple rescaling of the μ and σ axes and a redefinition of the variables the axes represent allows us to reinterpret Figure 4. The surface plot in the figure represents the DROP statistics for pairs $(\frac{\mu}{\gamma \cdot F}, \frac{\sigma}{\gamma \cdot F})$, and the set identifies the subset of $CLIFFSET$ for any parameterization in which $N = 100$ and $z = 2$.

$DROP$ is increasing in both N and z, and hence for larger values of either or both of these parameters the associated subset of $CLIFFSET$ is somewhat larger than the one identified in Figure 4. To give a sense of how sensitive the subset is to these variables we have calculated DROP statistics for $z = 3$ and for $N = 160$ for three points on the outer boundary of $CLIFFSET'$.

Table 5: DROP statistic for N=160 and z=3.

Threshold tuples $(\frac{\mu}{\gamma \cdot \mathbf{F}}, \frac{\sigma}{\gamma \cdot \mathbf{F}})$	**DROP**$(\mathbf{N = 160})$	**DROP**$(\mathbf{z = 3})$
(0.9125, 0.4088)	0.3781	0.3650
(0.9981, 0.2993)	0.3870	0.3763
(0.7825, 0.4856)	0.3968	0.3711

2.3 Generalizing the modeling assumptions

The geometry of the cliff is robust to a range of alternative assumptions in how p, q and g are determined. In Panels 1-3 of Figure 5 we present comparative visualizations of the cliff for three illustrative variations in the assumptions that determine p, q and g. In each case the solid line plots the baseline $E(v|R)$ cliff, and the dashed line plots the relevant comparison. In each case there is a distinct unruly regime, a compliant regime, a cliff-like transition between them, and a measured DROP exceeding our threshold.[7]

In Panel 1 we alter the technology of apprehension, which determines p. Two assumptions determine the objective probability of apprehension in the computable model: (i) exactly one unit of the enforcement resource, no more and no less, is required to investigate a violation; and, (ii) if a violation is investigated the violator is apprehended with probability γ. An alternative specification assumes that the probability that a violator is apprehended is continuous in the quantity of resources, r, devoted an investigation, $\gamma \cdot (1 - 1/\epsilon^r)$, where $\epsilon > 1$ and $0 < \gamma < 1$. This probability is increasing and concave in r and bounded above by γ. Given resources R, to maximize the expected number of apprehensions we must allocate $\frac{R}{v}$ to each violation. With this allocation, the probability that any violator is apprehended is $\gamma \cdot (1 - \frac{1}{\epsilon^{R/v}})$. In Panel 1 we assume $\epsilon = 8, \gamma = 0.8$

[7] In each case the value of DROP is sensitive to the assumed values of the parameters involved.

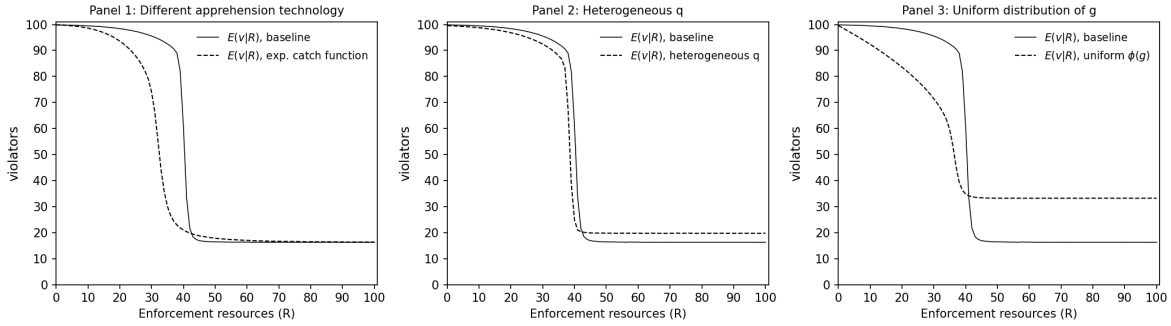

Figure 5: Robustness of the cliff, for variations in $P(v, R)$, q, and $\phi(g)$.

The code that produces Figure 5 can be found here:

- Matlab file(s): Figure 5

In Panel 2 we alter the specifications determining peoples' subjective probabilities of apprehension. The computable model assumes that all people share the same mapping from the z-history to their subjective probability of apprehension, $q_i^t = Q(zh^t)\ \forall\ i$. Alternatively, we assume $q_i^t = Q(zh^t) + \delta_i^t$, where the noise term, δ_i^t, is a person specific random draw from the uniform distribution with support $[-0.2, +0.2]$.

In Panel 3 we alter the process determining g_i^t. In the computable model, the g_i^ts are random draws from a normal distribution, $\phi(g_i^t)$. Alternatively, the g_i^ts are drawn from a uniform distribution with support $[0.2, 1.1]$.

2.4 The space of passive policies

Another issue concerns the robustness of our results to alternative levels of sanction. In the passive policies considered in the paper the sanction F was fixed, and R varied. We now examine passive policies in which both R and F vary. We restrict our attention to passive policies for which $R \in \{0, ..., N\}$ and $F \in \{0, 0.05, ..., 1.45, 1.5\}$. This results in a set of passive polices with $101 \cdot 31 = 3131$ elements. Figure 6 is a surface plot of $E(v|R, F)$ for this set of polices. Two cross-sections are highlighted. The first is produced by holding the baseline sanction constant and varying R. This reproduces the "cliff of positive feedback". The second is produced by holding enforcement spending constant and varying F. This produces another cliff of positive feedback.

The code that produces Figure 6 can be found here:

- Matlab file(s): Figure 6

Both cross-sections, as well as the surface itself, reveal the same regimes we saw in the paper: (i) the unruly regime consists of a plateau around the top edge of the figure; (ii) the compliant regime is the rectangular plateau at the bottom; and, (iii) the transitional regime is a cliff separating the two. The cliff of positive feedback is a pervasive feature of our dynamic process.

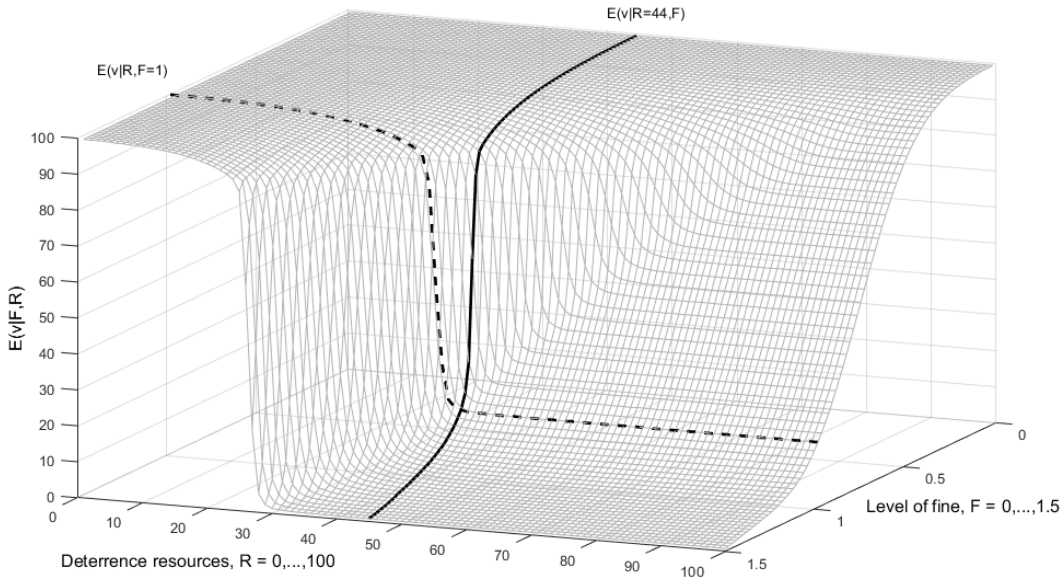

Figure 6: The space of passive policies.

3 Generality of deterrence policy analysis

In this concluding section we start by considering the generality and determinants of three central results for the baseline parameterization: (i) The reserve capacity rate with the optimal crackdown policy is higher than 60%, meaning that in a typical period only 40% of available enforcement resources are used to investigate violations; (ii) replacing the optimal passive policy with the optimal crackdown policy reduces the expected cost of ASB by 12%; and, (iii) replacing the optimal crackdown policy with the optimal refined crackdown policy reduces the expected cost of ASB by less than 1%. Naturally, these specific results, the reserve capacity rate and the measured reduction in costs, will differ from one parameterization to another. We show that they are driven primarily by RAT and $DROP$.

RAT is effectively the ratio of the external cost imposed on society by a violation to the cost of investigating a violation, so it is intuitive that it is an important determinant of these three numbers. As we have seen, managing ASB means managing positive feedback. Since $DROP$ is a measure of the potential for disruptive feedback it is intuitive that it too is important to understanding these results.

In Figures 7 and 8 we report the reserve capacity rate and the two cost reduction percentages for simulations in which N and z are fixed at their baseline values.[8] In these figures we are treating $DROP$ as something like a composite parameter. Given $N = 100$

[8] The estimates of optimal 2 bin and 3 bin dynamic policies reported in this section are generated using simpler versions of the directed search algorithms. The number of randomly seeded searches was reduced to 50 in each case implying that the errors in the estimation of the optimal policy will be slightly higher. But as we have observed, the space in which the optimum occurs is very flat and so there is very little difference between estimates in the neighborhood of the optimum. Reducing the number of searches increased computational efficiency at nearly zero cost to estimation precision.

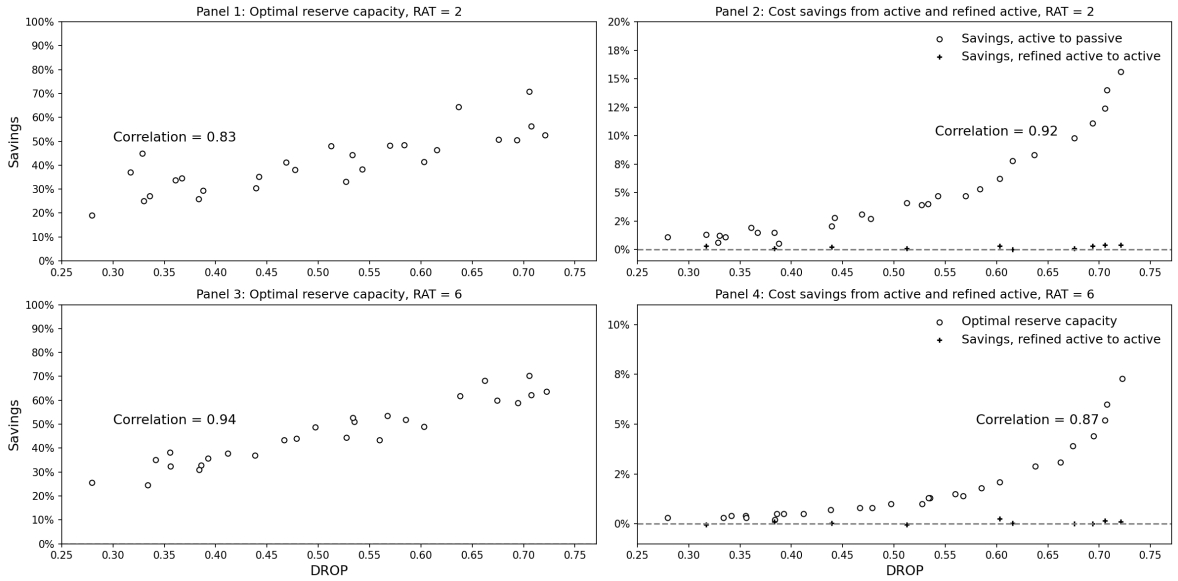

Figure 7: Reserve capacity and cost savings as a function of $DROP$.

and $z = 2$, as we saw in Section 2, $DROP$ is determined by $\frac{\mu}{\gamma \cdot F}$ and $\frac{\sigma}{\gamma \cdot F}$. In selecting the parameterizations used in constructing these figures, we used two criteria. First, that the distribution of $DROP$ be approximately uniform over the interval (0.275, 0.725). Second, subject to this constraint, that we include parameterizations distributed across the entire $(\frac{\mu}{\gamma \cdot F}, \frac{\sigma}{\gamma \cdot F})$ parameter space. For each parameterization we calculated reserve capacity and the two cost reduction numbers. Then, in Figures 7 and 8, we plot the values for our three central results for each parameterization with their values of $DROP$ and RAT.

The first thing to notice is the magnitude of the correlation coefficients reported in the four panels of Figure 7. The fact that they are relatively high is telling us that $DROP$ does function as a composite of the parameters that determine it, and the fact they are not 1 is telling us that the effects of variation in these parameters are not completely captured by $DROP$.

In Panels 1 and 3 of Figure 7, we report the reserve capacity rate for the optimal crackdown policy for a number of DROP values in the interval (0.25, 0.75). In Panel 1, RAT is fixed at 2 and in 3 it is fixed at 6. In both panels, reserve capacity is increasing in $DROP$, from a low of 20% up to high of 70%. In Panels 2 and 4 the circles represent the reduction in cost when the optimal passive policy is replaced with the optimal crackdown policy. In Panel 2 RAT is 2, and in 4 it is 6. The cost reduction numbers are positively associated with $DROP$ and negatively associated with RAT. The crosses in those panels represent the reduction in cost when the optimal crackdown policy is replaced with the optimal refined crackdown policy. These cost reduction numbers are negligible and uncorrelated with $DROP$.

In Panel 1 of Figure 8, we report the reserve capacity rate with the optimal crackdown policy as a function of RAT for three $DROP$ values. For all three values, there is massive discontinuity in the relationship between reserve capacity and RAT. (Given the coarse grid of RAT values, we have not identified any of the three discontinuities precisely.) When RAT

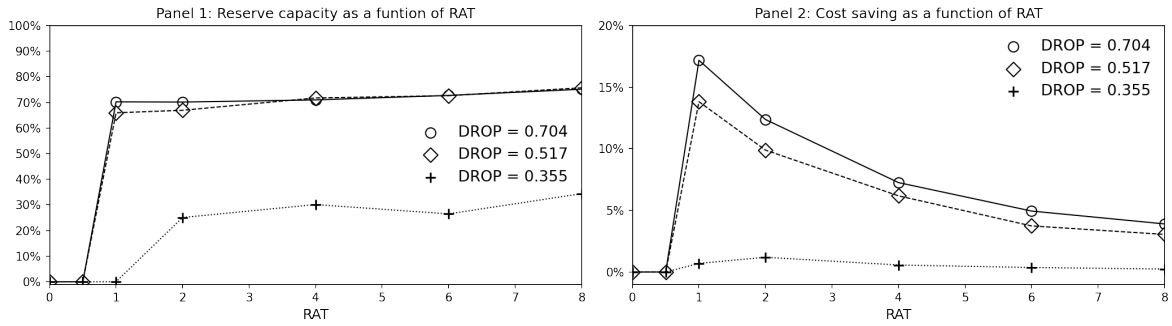

Figure 8: Reserve capacity and cost savings as a function of RAT.

is very small, we are in the spitting on the sidewalk case and the best enforcement policy is to do nothing. As RAT passes through the point of discontinuity the reserve capacity rate jumps up, and beyond the discontinuity it increases slowly with RAT. For the two larger $DROP$ values to the right of the discontinuity the rate exceeds 60% and for the lowest value it is much lower. In Panel 2, we report the reduction in cost when the optimal passive policy is replaced with the optimal crackdown policy as a function of RAT for the same three values of $DROP$. Again there is massive discontinuity in the relationship. When RAT is small do nothing is the optimal enforcement policy regardless of whether we are using a passive or crackdown policy, so the cost reduction is 0. Beyond the discontinuity the cost reduction is inversely related to RAT.

The results reported in Figures 7 and 8 are based on parameterizations in which $N = 100$ and $z = 2$. Because simulations with larger values of these parameters are very time intensive, we have run just a few of them. Based on those simulations it seems clear that the patterns seen in these figures and the magnitudes of the numbers are robust to increases in the values of the parameters.[9]

The code that produces Figure 7 can be found here:

- Matlab file(s): Figure 7

The code that produces Figure 8 can be found here:

- Matlab file(s): Figure 8

Our results support the following generalizations.

Proposition 2 *When the potential for positive feedback as measured by DROP is large: (i) to manage antisocial behavior in a cost effective way substantial reserve enforcement*

[9] Variations of $N = 160$ and $z = 3$ were run on two values each of $DROP \approx \{0.7, 0.4\}$ and $RAT = \{2, 10\}$. The pattern of cost savings for crackdown and refined crackdown was preserved. Costs savings for crackdown over passive increased slightly for both the larger N and z values, with the increases ranging from 0.1% to 1.8% over the baseline values with the maximum cost savings being 15.2%. The cost savings for refined crackdown was again negligible with the total savings ranging from 0% to 0.4%. The reserve capacity values were almost identical for each for each of the four cases. For example, for $DROP \approx 0.7$ and $RAT = 2$, reserve capacity = (0.4260, 0.4240, 0.4302) for the (N, z) tuples of (100, 2) (160, 2) and (100, 3) respectively.

capacity is required; (ii) the cost of antisocial behavior with the optimal crackdown policy is substantially lower than it is with the optimal passive policy; and, (iii) the cost of antisocial behavior with the optimal refined crackdown policy is negligibly less than it is with the optimal crackdown policy.

4 Empirical evidence for dynamic signatures

Here is some evidence of dynamic signatures reported in Section **??** from the the raw data.

Table 6: Autocorrelation of crime (US states, 1985-2021).

	Lagged crime rates		
	(1)	(2)	(3)
State	**1 lag**	**2 lags**	**3 lags**
AL	0.9004	0.7416	0.5766
AK	0.8518	0.6455	0.3309
AZ	0.9298	0.8316	0.7450
AR	0.7078	0.4593	0.2228
CA	0.9655	0.9033	0.8260
CO	0.9189	0.7897	0.6430
CT	0.9306	0.8371	0.7454
DE	0.8256	0.6577	0.4386
DC	0.9302	0.8239	0.6929
FL	0.9514	0.8892	0.8125
GA	0.9530	0.8700	0.7674
HI	0.1975	0.0476	-0.1575
ID	0.7784	0.6328	0.5143
IL	0.9591	0.8961	0.8275
IN	0.9054	0.7356	0.5366
IA	0.4974	0.3948	0.1860
KS	0.7677	0.4733	0.1455
KY	0.9102	0.7788	0.6229
LA	0.9335	0.8375	0.7031
ME	0.7318	0.6719	0.5403
MD	0.9311	0.8597	0.7885
MA	0.9290	0.8435	0.7451
MI	0.9340	0.8375	0.7610
MN	0.9085	0.7697	0.6151
MS	0.9203	0.8009	0.6454
MO	0.8980	0.7440	0.5966
MT	0.8619	0.7163	0.6038
NE	0.8580	0.6974	0.5155
NV	0.7358	0.4067	0.0974
NH	0.7955	0.6407	0.5062
NJ	0.9439	0.8745	0.7965
NM	0.8646	0.6909	0.5036
NY	0.9618	0.8993	0.8173
NC	0.9511	0.8682	0.7600
ND	0.9061	0.8242	0.7423
OH	0.9378	0.8455	0.7363
OK	0.8877	0.7139	0.5059
OR	0.9452	0.8800	0.8085
PA	0.8498	0.6956	0.5765
RI	0.9198	0.8548	0.7782
SC	0.9383	0.8505	0.7474
SD	0.8286	0.7488	0.6477
TN	0.7995	0.6076	0.4188
TX	0.9408	0.8485	0.7438
UT	0.9022	0.7647	0.6252
VT	0.7702	0.4618	0.1806
VA	0.9501	0.8786	0.7894
WA	0.9532	0.8922	0.8231
WV	0.8550	0.7128	0.6136
WI	0.6807	0.4681	0.3658
WY	0.7688	0.6408	0.4871

Table 7: Serial correlation of crime on policing (US states, 1985-2021).

	Lagged policing resources		
	(1)	(2)	(3)
State	**1 lag**	**2 lags**	**3 lags**
AL	-0.5037	-0.4815	-0.5890
AK	-0.0925	-0.2474	-0.3099
AZ	0.4066	0.3762	0.3761
AR	-0.6196	-0.6853	-0.7256
CA	-0.4231	-0.4668	-0.5279
CO	0.0995	0.0584	0.0079
CT	0.7422	0.7103	0.6614
DE	0.3070	0.2898	0.2935
DC	0.2370	0.1907	0.1227
FL	-0.2477	-0.2956	-0.4383
GA	-0.0053	-0.0520	-0.1388
HI	0.4231	0.4054	0.3566
ID	0.6057	0.6418	0.6639
IL	-0.6198	-0.6167	-0.6147
IN	-0.8648	-0.8989	-0.9163
IA	-0.7039	-0.7266	-0.7374
KS	-0.6725	-0.7074	-0.7326
KY	-0.5811	-0.6227	-0.6319
LA	-0.6145	-0.6723	-0.7566
ME	-0.8339	-0.8556	-0.8123
MD	-0.5174	-0.5747	-0.6215
MA	-0.6238	-0.6246	-0.6001
MI	0.7546	0.7378	0.7130
MN	-0.7286	-0.7413	-0.7431
MS	-0.4753	-0.5535	-0.6212
MO	-0.8247	-0.8645	-0.8803
MT	-0.7394	-0.7094	-0.6971
NE	-0.6263	-0.6746	-0.7143
NV	0.7254	0.6878	0.6607
NH	-0.4296	-0.4282	-0.4345
NJ	-0.1690	-0.0538	-0.0637
NM	-0.2783	-0.4779	-0.5886
NY	-0.3107	-0.3310	-0.3524
NC	-0.3217	-0.4521	-0.5632
ND	-0.5349	-0.5581	-0.5429
OH	-0.7487	-0.7159	-0.6708
OK	-0.6134	-0.6136	-0.5808
OR	-0.1400	-0.1500	-0.1603
PA	0.1120	-0.0420	-0.1684
RI	-0.6135	-0.6086	-0.5772
SC	-0.4784	-0.5614	-0.5938
SD	-0.6649	-0.6668	-0.6663
TN	-0.4090	-0.4538	-0.4759
TX	-0.2300	-0.3193	-0.3074
UT	0.1693	0.1249	0.0641
VT	-0.4667	-0.4646	-0.4222
VA	-0.7289	-0.7635	-0.7932
WA	0.3530	0.2962	0.2405
WV	-0.3404	-0.3435	-0.3632
WI	-0.2898	-0.2864	-0.3432
WY	0.0655	0.05607	0.1945
total -ve	**38**	**39**	**39**

Table 8: Autocorrelation of crime (CDN provinces, 1999-2020).

	Lagged policing resources		
	(1)	(2)	(3)
Province	**1 lag**	**2 lags**	**3 lags**
NL	0.6065	0.3016	0.0607
PEI	0.8656	0.6715	0.5324
NB	0.8465	0.6682	0.4827
NS	0.9097	0.7904	0.6832
QE	0.8751	0.7619	0.6495
ON	0.8530	0.7353	0.6251
MN	0.8826	0.7507	0.6186
SK	0.8775	0.7309	0.5895
AL	0.8083	0.5899	0.4159
BC	0.8943	0.7925	0.7160

Table 9: Serial correlation of crime on policing (CDN provinces, 1999-2020).

	Lagged policing resources		
	(1)	(2)	(3)
Province	**1 lag**	**2 lags**	**3 lags**
NL	0.3011	0.1366	-0.0131
PEI	0.2599	0.1209	-0.0696
NB	-0.0169	-0.1960	-0.3354
NS	-0.6996	-0.7716	-0.8407
QE	0.2145	0.1030	-0.0758
ON	-0.0225	-0.1591	-0.3070
MN	-0.4033	-0.6495	-0.7543
SK	-0.2348	-0.4459	-0.6042
AL	-0.6969	-0.7567	-0.7539
BC	-0.6105	-0.7227	-0.8180
total -ve	**7**	**7**	**10**