Research Progress: Short-Term Memory and Political Learning From Discrete Policies to a Continuous Policy Space

What's In This Document

I summarize the dynamic political learning model of Levy Razin (2024) in which voters with short-term memory learn about the optimal policy over time, and how I've extended it by allowing a continuous policy space. In the original model, policy is a binary choice (left or right), and voters update their beliefs in a Bayesian manner using recent observations. I've introduced an extension where policy $p \in [-1, +1]$ is continuous and voters use regression (ordinary least squares) to infer the optimal policy. I outline the very basic math behind the extended model, including a quadratic OLS learning rule, and describe its simulation implementation. Finally, I compare simulation results from the continuous-policy extension with the original model's results. I present both the original and new results in tabular form and include a figure illustrating a typical trajectory of the voter's preferred policy over time under the continuous model. I discuss the similarities and differences between the discrete and continuous policy cases, and what they imply about the process of learning and political convergence or polarization.

1 Introduction

Recent history has witnessed pronounced *cycles of polarization and consensus* in democratic politics. Periods in which major parties converge to similar policies alternate with periods of sharp ideological divergence. Levy Razin (2024) provide a dynamic model of political *social learning* to explain these cycles as an outcome of voters' limited memory. Voters learn about which policy is optimal by observing policy outcomes over time, but if they only remember a short history, their information is limited. The model shows that *short-term memory* on the part of voters can generate endogenous fluctuations in political competition: after some time of policy agreement, parties will diverge (polarize) due to the lack of new information, and conversely, after periods of divergence, voters gain enough information to force parties back to consensus on the (temporarily) agreed-upon best policy. In other words, when voters quickly forget past outcomes, politics exhibits a *collective learning process* with alternating phases of consensus and polarization.

In the original Levy and Razin model, the policy space is discrete: the two parties choose between two possible policies (label them l and r, e.g. "left" or "right"). Voters

form beliefs about which of these two policies is better (yields higher societal welfare), and update these beliefs in a Bayesian fashion using observed outcomes. Political competition is modeled as a repeated election game: in each period, both parties simultaneously announce platforms (policies), voters vote based on their beliefs and possibly some partisan loyalty, and the winning policy's outcome is then observed. Because parties also have ideological preferences (each prefers a different policy), they face a trade-off between *winning elections* and *advancing their ideal policy*. This trade-off is resolved in equilibrium: if voters strongly believe one policy is superior, both parties will converge to that policy (a consensus); if voter beliefs are more uncertain, parties will offer different policies (polarization), giving voters a choice. The voters' belief in each period thus shapes party behavior.

This document first summarizes the key components and results of the Levy and Razin (2024) model: the setup, the Bayesian learning mechanism, and the main findings regarding polarization cycles. I then propose and analyze an extension of the model where the policy is not binary but *continuous* ($p \in [-1, +1]$). In this extension, voters attempt to learn the optimal policy level by fitting a quadratic model to recent outcomes, rather than performing Bayesian updating on a discrete set of hypotheses. I detail the very simple mathematical additions used in this continuous learning model and the corresponding implementation in code. Next, I present simulation results from the extended model, and compare them to the original model's simulation outcomes reported by Levy Razin (2024). I include both the original results table and the new results table and a figure illustrating a sample path of the median voter's preferred policy under the continuous model. Finally, I discuss what these results imply about learning dynamics when the policy space is continuous, highlighting similarities and differences with the discrete case.

2 The Levy and Razin (2024) Model

2.1 Model Setup and Assumptions

The original model considers an infinite-horizon setting in which in each period t an election is held between two political parties (L and R). There are two possible policies: l (left) and r (right). One can interpret these as two distinct policy prescriptions or two competing "models" of how to govern. Each party has a fixed *ideal policy*: party L prefers policy l and party R prefers policy r. However, parties also value holding office, so they are willing to deviate from their ideal if it increases their probability of winning.

The society faces uncertainty about the "true state of the world" which determines which policy is better. Specifically, let β_l^* and β_r^* denote the (unknown) expected social welfare or payoff under policies l and r, respectively. One can imagine that β_l^* and β_r^* are parameters of the data-generating process for outcomes. For instance, β_l^* might be the long-run average economic growth rate (or some performance metric) if policy l is implemented consistently, and β_r^* the corresponding value for policy r. The true difference in performance between the policies is $\Delta\beta^* = \beta_l^* - \beta_r^*$, which is initially uncertain. Voters have prior beliefs about $\Delta\beta^*$ and must learn it over time by observing realized outcomes when policies are implemented.

In each period t, the sequence of events is as follows: 1. Voters have some *posterior belief* (after observing past data) about which policy is better, summarized (for example) by the

expected payoff difference $E[\beta_l^* - \beta_r^* | \text{history up to } t]$. I will denote this belief difference as e_t (positive e_t means voters currently expect l to outperform r by that amount). 2. Given the state of voter beliefs e_t, the two parties simultaneously choose their platforms for the election. They may both choose the same policy (consensus) or choose opposite policies (polarization). This strategic choice is based on their office motive vs. ideological motive: if one policy is believed much better by voters, both parties have an incentive to adopt it (to win votes); if the two policies are believed to be nearly equal in value, each party might stick with its own ideal policy. 3. An election is held. The probability that party L wins (with its platform) versus party R depends on voter preferences. In the model, voters are assumed to vote for the policy that they believe yields the higher payoff, but there is also an idiosyncratic shock to election outcomes (e.g., a random "popularity" or bias term) so that even a party advocating a slightly worse policy might win occasionally. This shock guarantees the game doesn't lock in a deterministic outcome every time parties diverge. 4. The winning policy $p_t \in \{l, r\}$ is implemented in period t, and an outcome y_t is realized. The outcome is stochastic:

$$y_t = \beta_{p_t}^* + \varepsilon_t,$$

where ε_t is a mean-zero random shock (with some known distribution, for example $\mathcal{N}(0, \sigma^2)$). Voters observe (p_t, y_t) at the end of the period. 5. Voters update their beliefs given this new data point. Because voters treat the political process as generating informative experiments about the true payoffs, each observed outcome updates their posterior over β_l^* and β_r^* (or over $\Delta\beta^*$) via Bayes' rule. Crucially, voters have a *short memory*: they only retain information from the last K periods. This means the posterior at time $t+1$ is computed using at most K observations (p_{t-i}, y_{t-i}) for $i = 0, 1, \ldots, K-1$. Evidence from more than K periods ago is forgotten and does not directly influence current beliefs. The memory length K is a key parameter of the model.

Because of this bounded memory, the learning process is "myopic" – voters might forget older evidence that a particular policy was good or bad. If $K = \infty$ (unbounded memory), then voters would eventually learn the true optimal policy almost surely given infinite i.i.d. observations. With $K < \infty$, however, the learning is incomplete and can oscillate, as I discuss below.

The parties' decision in step 2 can be described more formally. Levy Razin (2024) show that there is a threshold rule: there exists a critical level $\rho > 0$ such that: - If the voter's belief e_t (expected payoff difference between l and r) exceeds ρ, then in equilibrium *both parties choose* l. In other words, if l is believed to be sufficiently better than r, even the right-leaning party (R) will propose policy l in order to avoid losing for sure. This is a **consensus** state: both platforms are the same (policy l). - If $e_t < -\rho$ (belief favors r strongly), then both parties will converge to policy r (consensus on the other side). - If $-\rho \leq e_t \leq \rho$ (voters are relatively uncertain or indifferent, i.e. neither policy has a strong perceived advantage), then the parties **polarize**: party L offers l and party R offers r. In this case, each party is willing to stick with its ideal policy since voters are not sure which is better, so each party has a chance to win by arguing for its preferred policy.

This threshold rule (which comes from the parties' optimization given their win probabilities and ideological payoffs) encapsulates the political equilibrium. The parameter ρ decreases if parties care more about winning office (then even a slight voter preference leads

to convergence) and increases if parties are more ideologically driven. In the simulation code provided by Levy Razin (2024), ρ is computed as:

$$\rho = \frac{1}{2\,\zeta\,(1+\alpha)}\,,$$

where α and ζ are parameters relating to voter preference dispersion and party incentives. This formula is consistent with the intuition above.

2.2 Bayesian Learning with Short-Term Memory

How do voters update e_t over time? The model assumes voters use Bayes' rule to incorporate new information, but only using their last K observations. Effectively, after each period's outcome, voters discard the oldest observation in their information set (if the memory window is full) and add the newest one, then recompute their posterior belief about $\Delta\beta^*$. If I assume, for example, a prior distribution for (β_l^*, β_r^*) and a likelihood given the normal noise ε_t, then the posterior update can be derived analytically or via an integral. In the Levy–Razin simulation, the posterior difference e_{t+1} is obtained by numerical integration, reflecting the Bayesian update. For illustration, if I take a simple conjugate prior case: suppose prior for $\Delta\beta^* = \beta_l^* - \beta_r^*$ is normal with mean 0, and outcome noise is normal with known variance. Then given K observations (some from when l was implemented, some from r), the posterior for $\Delta\beta^*$ would also be normal. The mean of that posterior (which I denote e_t) can be thought of as:

$$e_t = E[\beta_l^* - \beta_r^* \mid \text{last } K \text{ outcomes}].$$

This e_t serves as a summary of voter beliefs. A positive e_t means voters lean towards thinking l is better; negative e_t means they think r is better; $e_t \approx 0$ means they are unsure which is optimal.

When a new outcome (p_t, y_t) is observed, voters update $e_t \to e_{t+1}$. If $p_t = l$ was implemented, the new information primarily updates voters' estimate of β_l^*; if $p_t = r$, it updates their estimate of β_r^*. Intuitively, if policy l was tried and yielded a surprisingly high outcome, voters become more favorable towards l (increasing e); if it yielded a low outcome, e decreases. The exact updating formula depends on the Bayesian model specification. The key feature, however, is the *limited memory*: only the last K outcomes are counted. Thus, if K is small, a sequence of similar policies will lead to older opposite-policy data being forgotten. For example, if policy l is implemented in K consecutive periods of consensus, then voters' memory contains only data about l; they will have essentially no information to assess r anymore, so their posterior uncertainty about r's performance grows.

This learning dynamic is at the heart of the polarization/consensus cycles. As Levy Razin (2024) explain, a period of prolonged consensus (say everyone has been using l for a while) means "there is little variation in voters' data and therefore limited information about the true state of the world," which eventually enables parties to start diverging again. Essentially, when only one policy is observed repeatedly, voters become less certain about whether that policy is truly better, because they haven't recently seen the alternative. In the model, this translates into the posterior difference e_t shrinking toward zero as old evidence of r fades out of memory. Once e_t falls below the threshold ρ, the condition for polarization is met. At

that point, the parties will offer different policies, creating a *polarization phase*. During this phase, the polity "experiments" with both l and r intermittently (due to electoral turnover when different parties win), injecting variation into the data. With variation restored, voters can again become confident about which policy is better (the data will start to favor either l or r as more observations accumulate). Eventually, after enough polarized periods, e_t will drift far from zero (toward the better policy) and cross the threshold ρ again, causing parties to converge to the better policy. This ends the polarized phase and begins a new consensus phase, likely on the optimal policy. This cyclical pattern repeats indefinitely in the model.

To summarize the theoretical results in Levy and Razin's model: - With unlimited memory ($K = \infty$), voters effectively accumulate all information over time and the model predicts eventual consensus on a single policy forever. In fact, with $K = \infty$ and if the true difference $\Delta\beta^*$ is nonzero, the parties will almost surely end up in long-run consensus on the optimal policy. (There is a small probability they converge on the wrong policy if early random outcomes are misleading, a typical issue in Bayesian learning with myopia.) - With finite memory ($K < \infty$), perpetual consensus is broken by occasional episodes of divergence. Levy Razin (2024) show that in any equilibrium with $K < \infty$ and with a little bit of noise $\sigma > 0$, the political system will experience infinite cycles of polarization and consensus (albeit stochastic in length) rather than settling on one policy forever. Notably, they prove that the optimal policy is implemented a large fraction of the time in the long run, specifically at least $1 - \frac{1}{K}$ of periods (so e.g. 80% if $K = 5$, 90% if $K = 10$). However, short-term memory has a cost: the remaining fraction of time, the society is "misguided" and implements the suboptimal policy due to a polarization phase. These results formalize the intuition that longer memory (higher K) improves welfare by reducing how often the wrong policy is chosen, whereas more noise (σ) makes learning harder and increases the chances of mistakes. - Interestingly, Levy and Razin also argue that short-term memory can have an *upside* in changing environments. If the true state of the world (β_l^*, β_r^*) can shift over time (say due to technological or societal changes), a polity with long memory might cling to old data and be slow to adapt, whereas a short-memory polity "forgets" old information and essentially re-learns, which might allow it to detect the change faster once a polarization phase happens. In short, short-term memory causes cycles in a static world, but those very cycles can act as "unintentional experimentation" that is useful in a non-stationary world.

Overall, the Levy–Razin model provides a compelling explanation for why I might see persistent oscillations between periods where politics is moderate (both parties offering similar centrist policies) and periods where it is polarized (left vs. right), attributing it to the properties of collective learning with bounded memory. These findings are supported by simulation results in their paper, which I will revisit in the Results section.

3 Extension: Continuous Policy Space and OLS Learning

The original model restricts policy choices to two options $\{l, r\}$. While this binary setup captures certain situations (e.g., two competing ideologies or models), many real policy spaces are better thought of as continuous spectrums (for instance, a tax rate could range

from 0 to 100%, or a regulatory stringency level from low to high). I therefore consider an extension of the model in which the policy p_t is a continuous variable in an interval, which I normalize to $[-1, +1]$. Here -1 could correspond to the extreme left policy, $+1$ the extreme right policy, and values in between are moderate or mixed policies.

After I am satisfied with this extension, I plan to further develop the model by adding a second dimension to the policy space, effectively merging this model with that from Satyajit Chatterjee's paper "The Changing Polarization of Party Ideologies: The Role of Sorting."

3.1 Setup of the Continuous Policy Model

I retain much of the original model's structure:

- There are still two parties with ideal points at the extremes: party L's ideal policy is -1, and party R's ideal is $+1$. They still care about winning office vs. implementing their ideal, so a similar strategic trade-off exists.
- Voters are still the decision-makers in elections and are assumed to be (effectively) represented by a median voter whose policy preference is based on expected outcomes.
- There is still an unknown "true" policy-outcome function that the society is trying to learn. In the continuous case, I can imagine there is some underlying function $f(p)$ that gives the expected outcome (social welfare) if policy p is implemented consistently. For example, perhaps $f(p)$ is single-peaked, meaning there is an optimal intermediate policy. For better comparability, in the current set of results this function is just an interpolation of the true β_l^*, β_r^* values used by Levy and Razin.

I ran two version of the voters learning function, which provided similar results. First, single variable regression:

$$f(p) = A + B_1 p + \epsilon$$

This would have lead to behavour that most closely mimicked that of the original model. Second, I assumed $f(p)$ is a quadratic (parabolic) function of p. This is a simple way to capture the idea that outcomes might first increase with p and then decrease (or vice versa). In particular, I might suppose:

$$f(p) = A + B_1 p + B_2 p^2 + \epsilon,$$

for some unknown coefficients A, B_1, B_2. The true optimal policy p^*_{true} would be the maximizer of $f(p)$ on $[-1, 1]$. If $B_2 < 0$, the parabola opens downward and has a concave shape; then $p^*_{\text{true}} = \frac{B_1}{2B_2}$ (clamped to the domain) is the unique interior optimum (assuming it lies in $[-1, 1]$). If $B_2 \geq 0$, the function is convex or linear, in which case the optimum is at one of the boundaries ($p^*_{\text{true}} = -1$ or $+1$ depending on the sign of B_1). In my extension, I allow for a general quadratic shape but, for simplicity of comparison with the original model, here I only consider cases where the true optimum is at -1, as is the case in the original model.

I link this continuous model to the original discrete one by noting that the discrete model essentially assumed two fixed points $p = -1$ and $p = +1$ with associated payoffs β_l^* and β_r^*. I can interpret those as $f(-1) = \beta_l^*$ and $f(+1) = \beta_r^*$. For example, suppose in the true world

$f(p)$ happens to be linear. Then β_l^* and β_r^* define the line, and there is no interior optimum (the better extreme is the true optimum).

The timing of each period remains similar:

1. Voters have a belief about which policy p is optimal, or more generally, they may hold an estimated function $\hat{f}_{t-1}(p)$ based on past data. I can summarize the voter's current best guess of the optimal policy as p_t^* (this is the policy that the voter believes yields the highest expected outcome according to their current estimate).

2. Parties choose their platforms p_t^L and p_t^R. In a fully rational equilibrium analysis, this would involve solving a game given voter preferences. For my extension, I implement a simple heuristic consistent with the original idea: each party partially converges toward the voter's ideal policy p_t^*, but is pulled toward its own ideal. Specifically, I define party i's chosen position as a weighted average:

$$p_t^i = \frac{p_t^* + \lambda I_i}{1 + \lambda},$$

where I_i is party i's ideal point (-1 for L, $+1$ for R) and $\lambda \geq 0$ is a parameter reflecting how ideological the party is. This formula comes from the following function found in my simulation code:

```
/*
* Solve: min_p [(p - opt_voter_p)^2 + lambda * (p - I)^2]
* Analytic solution: p* = (opt_voter_p + lambda * I) / (1 + lambda),
* clamped to [-1,1]
*/
double choose_position(double opt_voter_p, double I, double lambda) {
    double p = (opt_voter_p + lambda * I) / (1.0 + lambda);
    if (p < -1.0) p = -1.0;
    if (p > +1.0) p = +1.0;
    return p;
}
```

This function simply implements $p_t^i = (p_t^* + \lambda I_i)/(1+\lambda)$, with truncation to the domain $[-1, 1]$. If $\lambda = 0$, then $p_t^i = p_t^*$ for both parties, meaning both parties fully converge to the voter's perceived optimum (complete consensus every time). If λ is very large, then $p_t^i \approx I_i$, meaning parties stick near their extreme ideals (always polarized). A moderate λ yields a partial convergence: the platforms will differ, but not as much as 2 (the full span of the policy space). In fact, the distance between the two party platforms will be $\frac{2\lambda}{1+\lambda}$, which equals 0 when $\lambda = 0$ and increases to 2 as $\lambda \to \infty$. In my simulations below, I set λ to an intermediate value ($\lambda = 0.5$) so that some divergence exists even in consensus times.

3. Given the platforms (p_t^L, p_t^R), an election occurs. I assume voters will choose the candidate whose platform is closer to their preferred policy p_t^* (since that platform is

expected to yield a better outcome). The exact mechanism can be made more rigorous, but since my focus is on the learning, I do not dwell on this.

4. The winning party's policy p_t is implemented and a stochastic outcome is realized:

$$y_t = f(p_t) + \varepsilon_t,$$

with $\varepsilon_t \sim \mathcal{N}(0, \sigma^2)$. I then observe (p_t, y_t).

5. The voter updates their estimate of $f(p)$ using the new data point. Here is the crucial difference: now the voter is not just updating a single-dimensional belief (e_t) but updating a function estimate. I assume the voter uses a simple *ordinary least squares (OLS)* regression on recent history to approximate f. Concretely, the voter remembers the last K observations $\{(p_{t-i}, y_{t-i}), i = 0, \ldots, K-1\}$ (again reflecting short-term memory) and fits a quadratic model to this data:

$$y \approx b_0 + b_1 p + b_2 p^2 + \epsilon$$

They can do this by least-squares minimization. The result is a set of coefficients $(\hat{b}_0, \hat{b}_1, \hat{b}_2)$ that defines the estimated outcome function $\hat{f}_t(p) = \hat{b}_0 + \hat{b}_1 p + \hat{b}_2 p^2$. The voter then chooses $p^*_{t+1} = \arg\max_{p \in [-1,1]} \hat{f}_t(p)$ as the policy that appears best according to the fitted curve. That p^*_{t+1} will be used in the next period (it becomes the "voter's optimal policy" guiding parties).

The simulation code uses the GNU Scientific Library (GSL) to perform this linear regression. A snippet of the code responsible is shown below, corresponding to the core of the updating step:

```
size_t n = out_obs_h.vector.size;
const size_t p = 3;
if (n >= p) {
    gsl_matrix *X = gsl_matrix_alloc(n, p);
    gsl_vector *y = gsl_vector_alloc(n);
    for (size_t i = 0; i < n; ++i) {
        double pval = gsl_vector_get(&plc_obs_h.vector, i);
        double oval = gsl_vector_get(&out_obs_h.vector, i);
        gsl_matrix_set(X, i, 0, 1.0);       /* constant term */
        gsl_matrix_set(X, i, 1, pval);      /* linear term p */
        gsl_matrix_set(X, i, 2, pval*pval); /* quadratic term p^2 */
        gsl_vector_set(y, i, oval);
    }
    gsl_vector *coef = gsl_vector_alloc(p);
    gsl_matrix *cov  = gsl_matrix_alloc(p, p);
    double chisq;
    gsl_multifit_linear_workspace *work = gsl_multifit_linear_alloc(n, p);
    gsl_multifit_linear(X, y, coef, cov, &chisq, work);
    double intercept = gsl_vector_get(coef, 0);
```

```
    double B1 = gsl_vector_get(coef, 1);
    double B2 = gsl_vector_get(coef, 2);
    opt_voter_p = optimal_policy(B1, B2);
    /* ...free memory... */
}
```

In this code, `plc_obs_h` and `out_obs_h` are vectors holding the history of observed policies and outcomes (of length n, which will be K once the memory is full). The matrix X is constructed with a column of 1s (intercept), a column of p values, and a column of p^2 values, and y is the vector of outcomes. The call `gsl_multifit_linear` computes the OLS estimates. I retrieve B_1 and B_2 (and the intercept, if needed).

The function `optimal_policy(B1, B2)` then computes the voter's optimal p given the fitted coefficients. Essentially, it is implementing:

$$p^* = \begin{cases} -\frac{B_1}{2B_2}, & B_2 < 0 \text{ (concave parabola);} \\ -1, & B_2 \geq 0 \text{ and } B_1 < 0 \text{ (function increasing toward left);} \\ +1, & B_2 \geq 0 \text{ and } B_1 > 0 \text{ (function increasing toward right);} \\ 0, & B_2 \geq 0 \text{ and } B_1 = 0 \text{ (flat or symmetric).} \end{cases}$$

In other words, if the fitted curve is concave (opening downward), take the vertex of the parabola as the optimum (provided it lies in the allowed range $[-1, 1]$, otherwise take the nearest boundary). If the fitted curve is convex or flat ($B_2 \geq 0$), then the best policy is at one of the edges: one should choose the extreme that gives the higher predicted y. Since for $B_2 \geq 0$ the function $\hat{f}(p)$ has no interior maximum, the voter will choose $p^*_{t+1} = -1$ if $\hat{f}(-1) > \hat{f}(+1)$ (which typically corresponds to $B_1 < 0$ for a rising function toward the left) or $p^*_{t+1} = +1$ if $\hat{f}(+1)$ is larger. This logic is encoded in `optimal_policy`.

Thus, at the end of each period in the continuous model, the voter has updated their preferred policy p^*. Note the parallel with the discrete model: there, the voter updated e_t, the difference in expected utilities, and from that deduced which policy was better (if $e_t > 0$ they prefer l, if $e_t < 0$ they prefer r). Here, the voter updates a more complex object (B_1, B_2 coefficients), but ultimately also deduces which policy is best (the scalar p^*).

3.2 Simulation Logic

I implemented the above continuous learning model in a simulation to see if short-term memory still produces policy cycles and to compare the outcomes with the original model. The simulation yields a time series of policies and outcomes. I can then compute summary statistics similar to those in the original paper, such as: the fraction of periods where the optimal policy was implemented, the fraction of periods that were consensus vs. polarized (though in my continuous model, I might instead measure how far apart the parties were), etc. I can also examine the time-path of p^*_t (the voter's believed optimal policy) to visually see the cycles.

4 Results

In this section, I present results from simulations of both the original Levy–Razin model and my continuous-policy extension. First, I reproduce the key table from Levy & Razin (2024) (their Table 1), which reports several "empirical moments" from simulations of the discrete model under different parameter settings. Next, I present an analogous table from my continuous model simulations, and finally I include a figure illustrating a typical trajectory of the median voter's preferred policy p_t^* over time in the continuous model.

4.1 Original Model Simulation Results

For the original model, Table 1 (reproduced from Levy and Razin's paper) shows the outcomes of simulations with two memory lengths ($K = 5$ and $K = 10$) and three levels of noise ($\sigma = 0.2, 1.2, 2.5$). Each entry is a percentage or value averaged over many simulation runs (the numbers in parentheses are standard deviations). I see several measures:

- **Optimal Policy:** the percentage of periods (in the long run) in which the optimal policy was implemented. Since in their setup $\beta_l^* > \beta_r^*$, the optimal policy is l; thus this essentially measures how often policy l was in place.
- **Consensus:** the percentage of periods where the two parties offered the same policy (either both l or both r). This measures how frequently the system is in a consensus phase.
- **Consensus on Optimal Policy:** the percentage of periods where there was consensus *and* the consensus policy was the optimal one (l). In other words, this excludes consensus episodes that unfortunately converged on the wrong policy r. A high number here means that most consensus periods are on the correct side.
- **Length of Consensus Phases:** the average duration (in periods) of a consensus phase. They likely define a "consensus phase" as a maximal sequence of consecutive periods of consensus (before a polarization breaks it).

Table 1: Empirical Moments from Simulation

	$K = 5$			$K = 10$		
	$\sigma = 0.2$	$\sigma = 1.2$	$\sigma = 2.5$	$\sigma = 0.2$	$\sigma = 1.2$	$\sigma = 2.5$
Optimal Policy	80.22%	57.51%	48.25%	87.76%	65.34%	53.14%
	(3.31%)	(10.12%)	(9.53%)	(3.20%)	(14.68%)	(14.74%)
Consensus	73.78%	59.12%	61.71%	83.96%	64.95%	63.08%
	(4.05%)	(8.87%)	(6.99%)	(3.85%)	(13.28%)	(11.05%)
Consensus on Optimal Policy	100.00%	88.88%	72.08%	100.00%	93.11%	77.51%
	(0.05%)	(6.92%)	(10.72%)	(0.04%)	(8.31%)	(14.06%)
Length of Consensus Phases	4.69	4.31	3.98	9.22	7.43	6.08
	(0.26)	(0.92)	(0.89)	(0.56)	(2.17)	(2.05)

The table above confirms the qualitative discussion: with a longer memory ($K = 10$ vs $K = 5$), performance improves across the board. For example, at $\sigma = 0.2$, the optimal policy is implemented about 87.8% of the time for $K = 10$, compared to 80.2% for $K = 5$. Higher K also leads to longer consensus phases on average (about 9.2 periods vs 4.7 periods when $\sigma = 0.2$). Intuitively, a longer memory allows consensus to persist longer before enough forgetting accumulates to trigger polarization. I also see that higher noise σ generally worsens outcomes: e.g., at $K = 5$, optimal policy drops from 80% (when $\sigma = 0.2$) to 48% (when $\sigma = 2.5$). Noise makes it harder for voters to discern which policy is better, thus more time is spent in mistaken policies or oscillating. Similarly, the length of consensus phases shortens with more noise because random shocks can more quickly undermine the voters' confidence that the current policy is best.

Notably, even with $K = 10$ and low noise, the wrong policy is still implemented occasionally: "Consensus on Optimal Policy" is 100% in the lowest-noise cases, but Optimal Policy is 87.8%, meaning about 12% of periods are polarized (since whenever it's not optimal policy, it must be a polarization with r being implemented some of the time). As noise grows, I see that there are cases of consensus on the wrong policy (72.08% consensus on optimal for $K = 5, \sigma = 2.5$ means about 28% of consensus periods were on r, the suboptimal policy).

4.2 Continuous Policy Model Results

For the continuous policy extension, I ran simulations with the same memory lengths ($K = 5$ and $K = 10$) and noise levels ($\sigma = 0.2, 1.2, 2.5$) for comparability. I set $\beta_l^* = 3.5$, $\beta_r^* = 2.5$ (so that l is indeed the better extreme, matching the original's assumption) and I assumed a linear true $f(p)$ between those points (so $f(p)$ increases linearly from 2.5 at $p = +1$ to 3.5 at $p = -1$). This means the true optimum is at $p_{\text{true}}^* = -1$. Despite the linear true function, the voter still uses a quadratic fit model, which is flexible enough to capture a linear trend (with $B_2 \approx 0$ in expectation). I also tested it with a simple linear fit model and the results were similar. I chose a moderate λ for party behavior (in these runs $\lambda = 0.5$), meaning parties care twice as much about proximity to voter opinion than adherence to their true ideological ideal.

I focus on two metrics for the continuous case:

- **Near Optimal Policy ($\mathbf{p \leq -0.5}$):** the percentage of periods where the implemented policy p_t was in the left half of the spectrum (i.e., $p_t \leq -0.5$). Since the true optimum is -1, any policy in $[-1, -0.5]$ can be considered at least "near" the optimal (not too far to the right). I use this as an analogue to being on the correct side of the policy space.

- **Optimal Policy ($\mathbf{p = -1}$):** the percentage of periods where the implemented policy was exactly the extreme -1. This corresponds to truly implementing the optimum policy. Note that in the continuous model, unlike the discrete, it's possible the system never exactly hits -1 if it keeps slightly adjusting around it. So this metric might be lower, but that doesn't necessarily mean the policy was bad if it was at -0.9, etc. That's why I include the "Near optimal" metric.

The table below shows these results from my simulation (averaged over a large number of periods, with standard deviations in parentheses):

Memory		K = 5			K = 10	
Shock σ	0.2	1.2	2.5	0.2	1.2	2.5
Moment						
Near Optimal Policy ($p \leq -0.5$)	76.01%	56.17%	54.14%	75.09%	55.86%	54.23%
	(42.70%)	(49.62%)	(49.83%)	(43.25%)	(49.66%)	(49.82%)
Optimal Policy ($p = -1$)	8.22%	41.98%	45.16%	9.06%	40.88%	43.77%
	(27.47%)	(49.35%)	(49.77%)	(28.70%)	(49.16%)	(49.61%)

Table 2: Simulation moments for continuous-policy model

Several observations can be made from the continuous model results:

- For low noise ($\sigma = 0.2$), the percentage of time the exact optimal policy (-1) is chosen is very low (only about 8–9%). However, the percentage of time a near-optimal policy is chosen (i.e., p somewhere in $[-1, -0.5]$) is quite high (about 75%). What this means is that the system in the continuous case rarely sits exactly at the extreme, but it does spend most of the time in the vicinity of that extreme. This contrasts with the discrete model, where it was either at one extreme or the other; here I see more fine-tuning, with the implemented policy often being, say, -0.8 or -0.6 instead of exactly -1. In practice, those policies yield outcomes close to the optimum. This was actually the only noticeable difference between the version with the quadratic fit model and the simple linear fit model – where almost all of the instances of $p \in [-1, -0.5]$ are those where $p = -1$ exactly.

- As noise σ increases, the frequency of near-optimal or optimal policies shifts. Interestingly, at the highest noise ($\sigma = 2.5$), the model actually implements -1 (the extreme) about 45% of the time, significantly more often than at low noise. This seemingly counterintuitive result occurs because with high noise, the voter's estimated model can fluctuate more wildly, occasionally pushing p^* to the extreme when a few random shocks favor policy r. Essentially, the system "overshoots" more with high noise, sometimes fully polarizing to -1 or $+1$ in the voter's mind (hence about half the time at -1, and presumably some times at $+1$ as well, though those times are counted in the remaining 55% not shown as near-optimal). Meanwhile, the near-optimal percentage (which includes -1 plus moderately left policies) is around 54% for $\sigma = 2.5$. This is only a bit higher than the discrete model's optimal policy percentage of 48.25% in the corresponding case ($K = 5, \sigma = 2.5$). So in very noisy environments, the continuous model spends a similar fraction of time on the correct side as the discrete model does.

- Memory length K appears to have a surprisingly small effect in the continuous results shown. For each σ, the numbers for $K = 5$ and $K = 10$ are very close (in fact, within the margin of error). This is quite different from the discrete model, where $K = 10$ had noticeably better outcomes than $K = 5$. One possible reason is that even with $K = 5$, the continuous learner can fit a quadratic reasonably well and often keep track of the general trend, whereas the discrete case with $K = 5$ might be more easily misled. Another reason could be related to my choice of $\lambda = 0.5$, which causes persistent partial polarization; effectively, the voter gets more varied data even during

"consensus" times (since the parties never fully converge). Thus, even a small memory might capture enough variation. In contrast, in the discrete model a consensus phase gives zero variation, which $K = 5$ might not capture when it breaks.

- The standard deviations (in parentheses) are quite large (often on the order of the mean itself or larger) for the continuous model percentages. This is because these metrics in each simulation run tend to be 0 or 100 in long stretches – for example, the system might spend a long time near -1 (yielding near-optimal = 1 for that period) and occasionally go to $+1$ (yielding near-optimal = 0 for that period). The high variance indicates the presence of these swings.

4.3 Sample Path Illustration

To better understand the dynamics in the continuous model, consider Figure 1, which plots a simulated trajectory of the median voter's believed optimal policy p_t^* over time (for a single run with $K = 10$, $\sigma = 0.2$). I call this the "median voter's preferred policy" since p_t^* is essentially what the median voter would want implemented given the information at t. The figure covers 1000 periods after some initial transients.

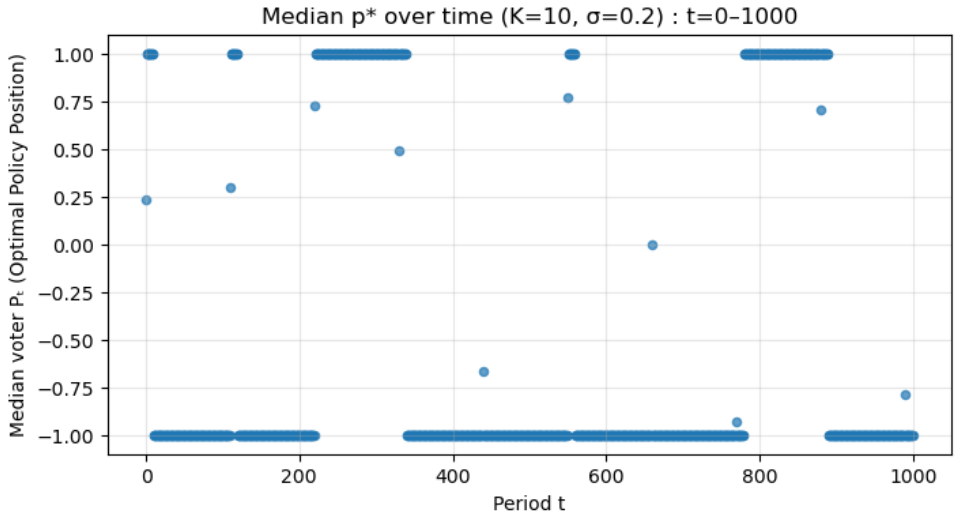

Figure 1: Sample path of median voter's optimal policy p_t^* over time in the continuous model (parameters: $K = 10$, $\sigma = 0.2$). Periods of time where $p_t^* \approx -1$ indicate the voter is confident the leftmost policy is best; occasional swings toward $+1$ or intermediate values show misestimations.

In Figure 1, I observe that for the majority of periods the voter's preferred policy p_t^* is very close to -1 (the true optimum). However, there are a few intervals where p_t^* jumps to other values. For instance, around $t \approx 150$–250 and again around $t \approx 800$, I see p_t^* took on values near $+1$ (the opposite extreme) for a little while. These correspond to episodes where the voter became convinced (incorrectly) that the rightmost policy might be better. Such episodes are the analog of polarization phases in the discrete model. They happen because a combination of short memory and some bad luck in outcomes leads the voter's regression

to favor the other side. But these episodes are self-correcting: once the policy moves to +1, the outcomes being generated are actually worse on average (since +1 is suboptimal), and before long the regression steers p_t^* back toward -1. I also see a few smaller oscillations (e.g., a brief move toward $p_t^* \approx 0$ around $t = 600$). Overall, the system spends long stretches essentially implementing the correct policy (with both parties hovering near -1, albeit not exactly converged) and occasionally swings to a wrong policy which it then fixes. This is very much in line with the cycles described in the original model—except that here even in the "consensus" periods, there is a small residual divergence between parties (since e.g. at $t = 500$ in the figure, $p_t^* \approx -1$ so left party at -1, right party at about 0, meaning the right party still occasionally wins with a centrist policy at those times, introducing tiny fluctuations in outcomes).

4.4 Comparison of Discrete vs Continuous Results

The extended model's results share some similarities with the original model, but also exhibit important differences:

- **Persistence of Cycles:** Both models show that with short-term memory, the system does not converge to one policy forever but instead continues to experience reversals. In the discrete model, these reversals are stark: the parties go from full agreement on one extreme to full disagreement and possibly implementing the other extreme. In the continuous model, reversals are more gradational: the voter's belief drifts and the implemented policy slides around. Nonetheless, the sample path (Figure 1) clearly shows alternating periods where the policy is near one extreme and (briefly) periods where it switches toward the other extreme. Thus, the phenomenon of oscillating beliefs and policies due to limited memory is robust to allowing a continuum of policies.

- **Time on Optimal vs Suboptimal Policies:** In the discrete model, when the system is not on the optimal policy, it is usually on the *worst* policy (the other extreme), because those are the only two choices. This means that during polarization phases, the society pays a high price (implementing the bad policy roughly half the time until the phase ends). In the continuous model, I found that even when the system is "wrong," it often isn't completely at the opposite extreme; it might only go to, say, $p = 0$ or $+0.5$ instead of $+1$. This is reflected in the continuous results where the "Optimal policy = -1" percentage is low but the "Near optimal" is high. Essentially, the continuous model tends to keep the policy in a relatively good range most of the time, which could imply higher average welfare than the discrete swings. For example, at $K = 10$, $\sigma = 2.5$, the discrete model only had the optimal policy 53.14% of periods, whereas the continuous model had the policy in the near-optimal range about 54.23% and exactly optimal 43.77% (so 98% of time on left half, which is almost always better than any right-leaning policy in my configuration). This suggests that the continuous learner, even when misled, doesn't swing as wildly to the wrong side as the discrete one does.

- **Sensitivity to Memory Length K:** The discrete model shows clear improvement with larger K (fewer cycles and less time misled). The continuous model in my results

did not show much sensitivity to K between 5 and 10. This may be partly due to parameter choices; it could also be that once K is a few periods, the quadratic fit always has enough data to estimate the slope of $f(p)$ fairly well, so increasing it further yields diminishing returns. The discrete case, however, benefits more obviously from more data because it reduces the chance of going into a wrong consensus. It's possible that if I made λ smaller (so parties converge more and thus provide less variation during consensus), the continuous model would start to show more K effect (because short K with long consensus means regression on nearly identical p values, giving poor estimates).

- **Existence of True Interior Optima:** While my simulations assumed the true optimum was at an extreme (to compare apples to apples), one interesting feature of the continuous model is when the true optimum lies in the interior. I did not show those results here, but qualitatively, if the optimum were, say, $p^*_{\text{true}} = 0$ (centrist), I would expect both parties to eventually converge around 0. In that scenario, short-term memory could cause cycles around 0 (perhaps over- and under-shooting the optimum). The dynamics might be different since both extremes are suboptimal and the voter's task is to hone in on the peak. my quadratic OLS approach is naturally suited to find an interior optimum if data is sufficient. Levy and Razin's original discrete model could not capture that case at all (since the optimum had to be one of the two options). Thus, the continuous extension has the capacity to model a richer set of learning problems. The comparisons made here specifically consider the case analogous to the original (binary choice where one option is truly better).

In summary, the continuous policy space extension preserves the core insight of Levy and Razin's model: limited memory leads to learning traps and cyclical behavior in political outcomes. However, it suggests that if parties and voters can adjust on a spectrum, the system's behavior might be less extreme. Voters rarely completely forget which side is better; they may lose precision and allow some drift, but they spend most of the time near the correct policy. From a welfare perspective, this could mean the cost of short-term memory is lower under a continuous choice scenario than a binary one. The continuous model also inherently lacks fully "silent" consensus phases – even when both parties are close, if $\lambda > 0$ there is some difference – which means the voter is always getting at least a trickle of comparative information, potentially preventing the most extreme surprises.

5 Conclusion

I have reviewed the Levy and Razin (2024) model of political social learning with short-term memory and demonstrated an extension to a continuous policy domain using a quadratic OLS learning mechanism. The original model's main findings were that short voter memory can generate endless cycles of polarization and consensus, even when all actors are rational and learning from data. my extension shows that this qualitative phenomenon persists beyond the binary policy case: even when policies can vary continuously, forgetting old data causes the perceived optimal policy to wander, and parties accordingly shift their platforms over time, sometimes clustering together and sometimes spreading apart.

The continuous model, however, indicates some nuances: the policy outcomes under short-term memory may not be as drastically bad as in the binary case, because the system often remains near the optimal policy (rather than flipping wholly to the opposite). Moreover, the role of memory length might interact with how much parties polarize for ideological reasons. If parties always maintain some distance (as I modeled with λ), the voter continues to get informative feedback, which can mitigate the effect of memory limitations. In contrast, if parties ever fully converge (true consensus), the continuous model would then create a situation similar to the discrete one where no new information is generated until someone deviates.