Anoma Research Topics | TECHNICAL REPORT

# Slow Games: Policy Enforcement under Uncertainty

**D Reusche[a], Christopher Goes[a], and Nicolas Della Penna[a]**

[a]Heliax AG

* *E-Mail: d@heliax.dev*

## Abstract

Motivated by decentralized permissionless protocols that are ultimately backed by social consensus, which can only perceive and act much slower than the service provisioning, we study what we term a Slow Game; a type of principal-agent problem, in which the agent acts as operator of a service and the principal as a regulator, which sets and attempts to enforce policies on the service being provided. The regulator is slower acting and measuring than the operator, which introduces uncertainty depending on the difference in speed. In this publication we introduce a framework inspired by lossy compression problems to model this type of game, as well as present results from simulations of a minimal example.

**Keywords:** mechanism design ; distributed systems ; principal-agent problem ; lossy compression

(Received September 14, 2024; Version: September 15, 2024)

## Contents

# 1. Introduction

Motivated by decentralized permissionless protocols that are ultimately backed by social consensus, which can only perceive and act much slower than the service provisioning, we study what we term a Slow Game; a type of principal-agent problem, in which the agent acts as an operator of a service and the principal as a regulator, which sets and attempts to enforce policies on the service being provided. The regulator is slower acting and measuring than the operator, which introduces uncertainty depending on the difference in speed. In this publication, we introduce a framework inspired by lossy compression problems to model this type of game, as well as present results from simulations of a minimal example.

Even though this work was inspired by systems where loss is induced by speed differences, it should apply to other setups in which a principal makes lossy observations of a world state influenced by an agent.

## 1.1. Conceptual framework

An instance of the *slow game* consists of, at minimum:

1. A *fast agent* $f$ (which might be a coordinated group) taking actions. The identity of the fast agent, their action space, and the costs/rewards of taking particular actions are specific to each instance.

2. A *slow agent* $s$ (which might be a coordinated group) taking measurements $m$. The identity of the slow agent, the measurements which can be taken, how frequently they can be taken, and how much they cost to take are specific to each instance.

3. A *world model* $w$ (which may or may not be fully known), which determines how the actions taken by the fast agent affect the measurements

taken by the slow agent (often over time). The nature of the world model (and how much of it is known) is specific to each instance.

4. A *regulatory mechanism r* through which the slow agent can reward or punish the fast agent, depending on the measurements which they take over time. The nature of the possible rewards and punishments is specific to each instance.

5. A *target world profile t* chosen by the slow agent (often changing over time). This target profile may include actions taken by the fast agents, measurements taken by the slow agents, or in-between (inferable) variables of the world state. The type of the target world profile is specific to each instance, and the value is typically an input to the system over time.

The characteristic questions for a slow game instance are:

Given the action space and costs/rewards of the fast agent, the measurement space, frequencies, and costs of the slow agent, the (possibly uncertain) world model, and the available regulatory mechanism:

1. Can a policy $p$ be crafted which will achieve the target world profile in incentive-compatible equilibrium?

2. What is that policy?

3. What is the deviation between the reward profile of the actions which best maximize the target world profile and the reward profile of the actions which best maximize the fast agent's returns? This could be called something like slack (or extractable value - this is a sort of generalized MEV).

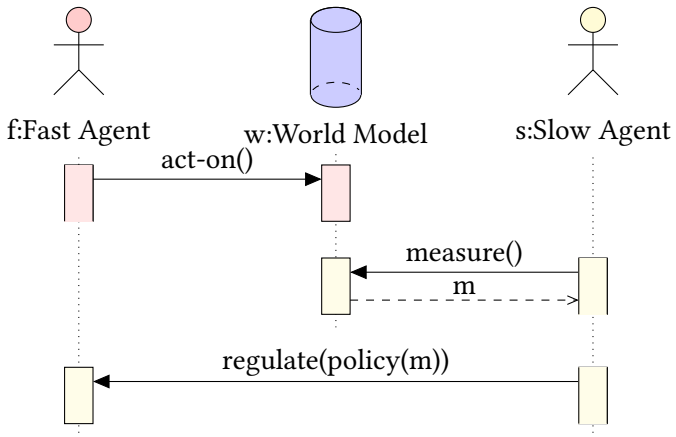


**Figure 1.** Slow Game Framework.

### 1.2. Examples

Here are some slow game examples, and how they instantiate each of these variables:

#### 1.2.1. Controller selection in Anoma

- Fast agent: controller in question, who can choose what fees to charge, and which transactions to possibly censor. The controller's reward is the fees paid, and possibly side rewards (bribes) for censorship.
- Slow agent: users submitting transactions to the controller in question, who can measure the fees charged, and can measure over time whether particular transactions are being censored.
- World model: fees are directly measurable; censorship is probabilistically measurable over time (since we also assume unreliable network conditions).
- Regulatory mechanism: users can decide whether to pay fees or not, and they can switch controllers, which reduces future rewards for the controller to zero.
- Target world profile: controller charges fees not more than a fixed margin above its operating costs, and what would be needed to clear the market, and controller does not censor transactions.

#### 1.2.2. Solver selection in Anoma

- Fast agent: Solver in question, who can choose to accept or not accept particular intents and to exploit slack (price differences between intents) or to return slack back to users.
- Slow agent: Users submitting intents to the solver, who can measure (over time and by comparing with each other) whether the solver is censoring intents and how much slack is being returned to users.
- World model: slack (MEV) return and censorship are probabilistically measurable over time (since we also assume unreliable network conditions).
- Regulatory mechanism: users can decide whether to keep sending intents to this particular solver, which reduces future rewards for the solver to zero.
- Target world profile: solver exploits slack not more than a fixed margin above its operating costs and does not censor intents.

#### 1.2.3. Delegated governance systems

- Fast agent: governance delegates, who can make particular decisions more for their own benefit or more for the benefit of a public (slow agent).
- Slow agent: voters, who can measure which decisions are made, or at least their impacts.
- World model: decisions made impact the state of the world (very general).
- Regulatory mechanism: varies, often voting out particular delegates on a periodic basis, sometimes also emergency referenda.
- Target world profile: general happiness and stability.

## 2. Relation to the literature

### 2.1. Foundations

#### Principal agent problems under uncertainty

The principal agent problem, introduced by [Ros73] describes the problem of deriving a fee schedule, s.t. an agent acting on a world state is incentivised to choose actions that lead to outcomes desired by a principal. The treatment concerns the perfect information setting, but acknowledges that principal agent relationships usually happen under information asymmetry.

#### Application of lossy compression approach

To derive methods on how to take into account measurement errors of the world state into reward structures, we utilize an approach from the field lossy compression, founded by [S+59]. The work we are building on is [BM19] from which we take the concept of *perceptual divergence*. Perceptual divergence enables us to derive how lenient the principal should be in enforcement, given a known measurement error.

#### Regret Formulation

To determine incentive structures, we use an approach based on *external regret*, in which the loss of a chosen action is compared to the loss of alternative actions in hindsight, regarding a chosen policy. This notion was introduced by [Han57] and [BM07] provide a generalized definition, the special case of which we use.

## 2.2. Related work

### Principal agent problems with uncertainty

[GW11] analyse principal agent problems under different kinds of uncertainty from a stochastic programming perspective. Section 3.3. drafts how to treat imperfect knowledge of the principal about the agent, but perfect knowledge about all other parameters, as well as a conjecture about the agents decision problem. The solution approach assumes connectedness and convexity of the solution set and solves problem formulations with stochastic quasi gradient methods. Our simulation based approach does not require a conjecture about the agents decision problem, only observations of outcomes, does not require the solving of stochastic programming problems, but the computation of wasserstein distances, and relies on knowledge assumptions specified in Section 3.1.1. We assume that observations of outcomes can be represented as discrete nonparametric distributions over a common metric space. The speed games described in our Section Section 3.3, implement a specific type of monitoring as described in Section 3.3.2. of the above paper, and are intended to reveal speed information.

### Prior free mechanisms

[CHJ20] started a line of work that does away with the assumption that principal and agent have a common prior over the world state, with both parties learning about the state over time.

The authors introduce a refinement of internal regret (where best-in-hindsight actions are determined regarding actions under a specific policy) called counterfactual internal regret (CIR), in which regret for a given action is computed regarding best-in-hindsight actions across all counterfactual policies. This is motivated by the behavioral assumption, that an agent who has access to private information that can be utilized under one policy, should also utilize it under all other policies, independent of the policy actually chosen by a mechanism.

Using CIR, the authors describe non-responsive, variable policy mechanisms for iterated games, where a policy is chosen in each iteration. They obtain regret bounds for the principal in symmetric information settings (Theorem 1), as well settings in which the agent possesses an informational advantage(Theorem 3). The bounds correlate regret of the principal with robustness of a policy against private information the agent possesses. A derivation is provided to transform mechanisms from symmetric information settings into the above.

We assume that the behavioral assumption applies to the settings of interest for slow games. Since we describe a non-responsive, fixed policy mechanism for non-iterated games with common priors and compute an example policy in a setting in which the principal and agent are equally well informed,

our setting should be a special case of this work[1] and Theorem 1 should provide a compatible regret bound.

[CRS23] improves on this work, by improving learning efficiency and relaxing assumptions on agent behavior.

**Solvers in intent-based markets**

The first part of [CKPD24] describes a concrete model, leveraging auction theoretical results to derive theorems about solver competition and user welfare under specific existing mechanisms with varying environmental setups. The authors provide results for a wide range of settings, including congestion cost and no-congestion cost, as well as for exponential, uniform and pareto price distributions. The latter heavy tailed distribution can not be treated reliably by our model (seeSection 3.4).

The second part treats an approach based on convex optimization, which provides more flexibility. It is used to corroborate the theoretical results, by solving instances of the formulated problem.

In contrast to the authors analyzing existing systems and giving strong results, our work introduces an observation based approach to structure incentives for fast service providers using slow regulators, but is restricted to a simple simulated model.

## 3. Lossy compression model

### 3.1. Model description

We assume that the difference in speed between operator and regulator leads to only lossy observations of operator actions (or outcomes thereof) being possible on the regulator side: We call this difference in speed the **speed factor**, the loss induced by its **dropout**.

**Example 1.** If the operator acts ten times within an interval, but the regulator can only measure two times, only 20% of the signal can be observed, the other 80% being dropout. The speed factor of the regulator, in this case, is 0.2.

Since we are interested in a quantitative analysis of how feasible it is for regulators to detect out-of-policy behaviours enacted by operators under uncertainty as described above, we take inspiration from lossy compression research, especially the concept of **perceptual quality**[2]:

[1] Full proof pending.

[2] [BM19]

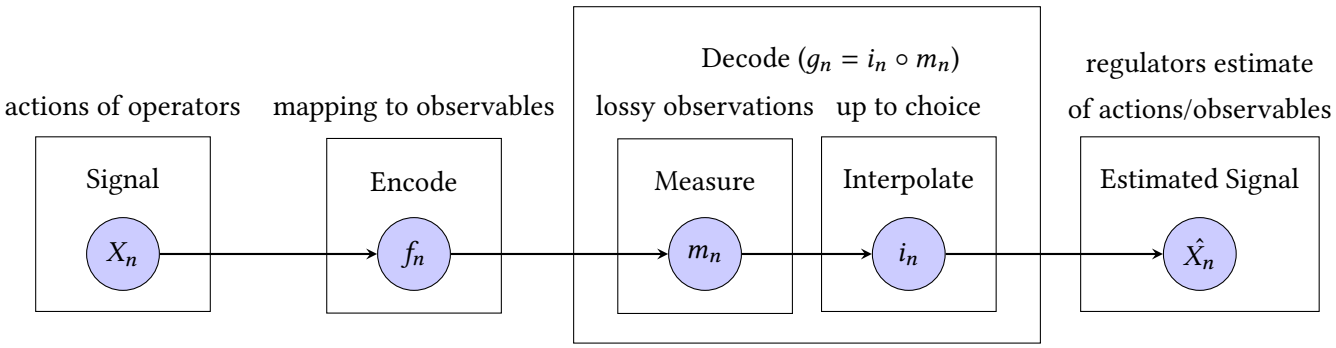


**Figure 2.** Lossy compression model for operator ⇔ regulator interactions.

For Figure 2, $d(X_n, \hat{X}_n)$ describes the **perceptual divergence** of a signal and its estimate, with $d = 0$ meaning that signal and estimate are fully equal. Using the Wasserstein metric[3] as a divergence measure, we can quantify the similarity between the distribution pairs of a signal and its estimate.

The *perceptual quality* of an estimate quantifies how likely it seems to correspond to a valid signal, in our case, a set of in-policy actions, independent of what the signal actually was.

### 3.1.1. Game formulation and knowledge requirements

To build the bridge from perceptual quality to the slow game problem, we can rephrase the last statement as follows: *How much could an operator cheat, while still producing a signal of which the estimate convinces the regulator of in-policy behaviour, given the regulators' lossy observations?*

To answer this question, we need to know the following:

**1) The speed factor at which observations happen.** See Section 3.3.

**2) A baseline distribution implementing in-policy behaviour.** This is used as a signal to compute estimates with simulated, slowness-induced loss and, in turn, the slowness-induced measurement error (*pure error*). Requirements for this are knowledge of the policy, as well as distributions the actions are drawn from (e.g. preferences or constraints of participants). Given these, we can, e.g. derive closed-form models or produce empirical distributions via simulation.

**3) Optionally, information about the cheating mechanism of the operator.** More specifically, information about the conflation of pure error and cheating mechanism for out-of-policy behaviours at known speed factors.

**Remark 2.** In many cases, the operator will have access to the regulators' knowledge, but more rarely the reverse be the case: A service provider can

[3] We use the Wasserstein metric with square Euclidean distance, instead of, e.g. KL-Divergence, because it is a proper metric, i.e. gives us interpretable values everywhere.

be their own user easily, but a user not always the provider of a service they are using.

We now introduce notation for these relevant types of signals[4]: Let $G_n$ be a baseline of "good" in-policy behaviour, $B_n$ an example of "bad" out-of-policy behaviour, $S_n$ some observed sample behaviour, with $\mathbb{S} = \{s \in \mathbb{Q} \mid 0 \leq s \leq 1\}$ being a family of speed factors $s$ and $d_s(\cdot, \cdot)$ the divergence measure of signal to estimate at a given speed factor $s$.

For a given speed factor $s$, we define the following metrics:

- $e_s(G_n) = d_s(G_n, \hat{G}_n)$ the **pure (slowness induced) error** of observations by the regulator. It tells us how close the estimate of a known good signal is to the signal itself.
- $c_s(B_n) = d_s(B_n, \hat{B}_n)$ the **cheating prior**. It tells us how close the estimate of a *specific* bad signal is to the signal itself, *including* interactions of the cheating mechanism with the pure error (the above mentioned conflation).
- $o_s(G_n, \hat{S}_n) = d_s(G_n, \hat{S}_n)$ the **observed divergence (from baseline)**. It gives a distance between the estimate of some observation from the estimate of a known good signal.
- $x_s(G_n, \hat{S}_n) = o_s(G_n, \hat{S}_n) - e_s(G_n)$ the **excess divergence**. It tells us how much of the observed divergence is not explained by pure error.

**Remark 3.** For ease of exposition, we look at observables $O_n = f(X_n)$, instead of the actions/signal $X_n$. In general, actions might not be observable at all, i.e., there is never access to signal samples. Thus, policies should be defined over observables $O_n = f(X_n)$, and estimates $\widehat{O_n}$ should be computed accordingly, unless the mapping between an observable and the latent variable modelling the signal is clear and policing the actions directly is desirable.

## 3.2. Crafting incentive structures

### 3.2.1. Reward mechanism

Since we want to incentivize in-policy behaviour, we need to define a reward/punishment mechanism to achieve that.

For example, assuming some base reward $R_b$ and operating cost $C_o$, we could try to compute a weighting factor $w$, which depends on how far we deem the operators behaviour to be away from in-policy behaviour while taking the uncertainty of our measurements at a given speed factor $s$ into account.

[4] We assume that an estimate can be computed for any signal that is available, but not the other way around.

To do that, we can use the measures for pure error, cheating prior and excess divergence from above to define *payoff weights*. In general, we subtract the excess divergence from the respective prior:

- In case we only know a baseline $G_n$:

$$w_s^G(G_n, \hat{S}_n) = e_s(G_n) - x_s(G_n, \hat{S}_n) \tag{1}$$

- In case we also know the cheating prior $B_n$:

$$w_s^B(G_n, B_n, \hat{S}_n) = c_s(B_n) - x_s(G_n, \hat{S}_n) \tag{2}$$

**Remark 4.** The *pure error* can be seen as a prior with no information about conflation with cheating distributions

This gives us reward weights $w_s$, which we can use directly in our payoff function. Then *payoff* for $S_n$ at speed factor $s$, derived from a good baseline is: $p_s^G(\hat{S}_n) = (R_b - C_o) \cdot w_s^G(G_n, \hat{S}_n)$. If cheating priors are available: $p_s^B(\hat{S}_n) = (R_b - C_o) \cdot w_s^B(G_n, B_n, \hat{S}_n)$. When not explicitly denoted, $p_s$ can be either $p_s^G$ or $p_s^B$.

### 3.2.2. Regret formulation

To check how well we incentivize in-policy behaviour with the payoff function from above, we calculate *external regret*[5]. for all parameter sets of the cheating mechanism, which are simulated per speed factor. E.g. if the cheating mechanism samples from a binomial distribution $B(10, c)$ with $c \in \{0.1, 0.2, ..., 1\}$, we receive for each $c_i$ a different corresponding $\hat{S}_n^i$. In our simulation, lower values for $c$ mean less cheating, with $c = 0$ being no cheating at all.

**Remark 5.** We assume the full information setting, in which we know payoff results for all choices of $c$. This has nothing to do with our measurements, it purely regards whether or not we know outcomes of alternative actions for regret calculation.

Regret for a specific action choosing $c$ then is the maximum pairwise difference in payoffs, holding fixed the payoff of the estimate $\hat{S}_n^j$ corresponding to $c_j$:

$$r_s(c_j) = \max_{\forall c_i} \left\{ p_s(\hat{S}_n^i) - p_s(\hat{S}_n^j) \right\}. \tag{3}$$

So, if we want the dominant strategy to be in-policy behaviour, regret should be minimized at $r_s(0)$ for any given $s$.

[5] As defined in [BM07], Section 2.

**Remark 6.** When using weights $w_s$, we can observe empirically in the experiment explained below, that in-policy behaviour minimizes regret for the operator, with reward being positive. Further work will need to show if this generalizes.

### 3.3. Speed Games

To determine the speed factor between regulator and operator, another game can be played, which we sketch here: Since a lower ratio of regulator: operator speed leads to more leniency of the regulator in our setting, the incentive of the operator is to convince the regulator of as high a speed as possible.

Assuming the regulator incurs some cost $c(f)$ for measuring at frequency $f$, the operator could offer (a part of) this cost to compensate the regulator for the process of proving their capability to operate at $f$.

Actual operation after the proof could take place at a lower frequency, but depending on measurement protocols, the regulator might detect that and adjust the speed factor in its models, plus some additional punishment, e.g. in case some operating speed is agreed on.

To access a wider range of trade-offs between measurement cost and strength of deterrence, the regulator can, e.g. randomize the measurement frequency.

### 3.4. Interpolation and heavy tails

Since dropout leaves us with incomplete data, we have the choice of interpolation scheme, e.g. replacing missing values with the mean of the interval, or using linear, polynomial or spline interpolation.

This has implications for which types of policies are feasible to (approximately) enforce: If payoff for defection is distributed in subgaussian fashion, i.e. "small" amounts of value can be extracted in a lot of events, interpolation will introduce tolerable error. If defection payoff is distributed in very heavy-tailed ways, i.e. a lot of value can be extracted in very rare events, interpolation error, is potentially very large.

Because of this, setups with subgaussian defection payoff are preferable. E.g. is certain choices of constraints for the system can be chosen that smooth out the distributions, that is preferable.

## 4. Example: Two-player thermostat

Let us now work out (and implement simulations for) [6] a minimal example of a slow game using the above approach. For that, we pick a two-player thermostat:

[6] The implementation can be found at https://github.com/anoma/slow-game-research.

### 4.1. Game Model

We have the following roles and objects:

- **Outside**, which has a fluctuating temperature (drawn once per timestep from a discrete uniform distribution $\mathcal{U}(10, 32)$) and influences the temperature of a room.
- A **room**, which is supposed to be kept within a certain range of temperature.
- An **operator** which
  - heats and cools the room to control its temperature.
  - tries to maximize its reward for the service provided (i.e. is a profit-maximizing actor), using a stochastic cheating mechanism to cool or heat slightly less than necessary (drawing from a binomial distribution $D_c = B(n, p)$).
- A **regulator**, which
  - sets the policy for the temperature bounds of the room. Here, the range is $[18, 25]$.
  - tries to verify policy adherence of the operator.
  - rewards or punishes operator depending on the degree of adherence to policy.

The reward is computed by setting a heating/cooling budget $R_b$ for a period with $TS$ timesteps, and giving all unspent budget to the operator as a base reward. Heating or cooling by one degree costs one unit of the budget.

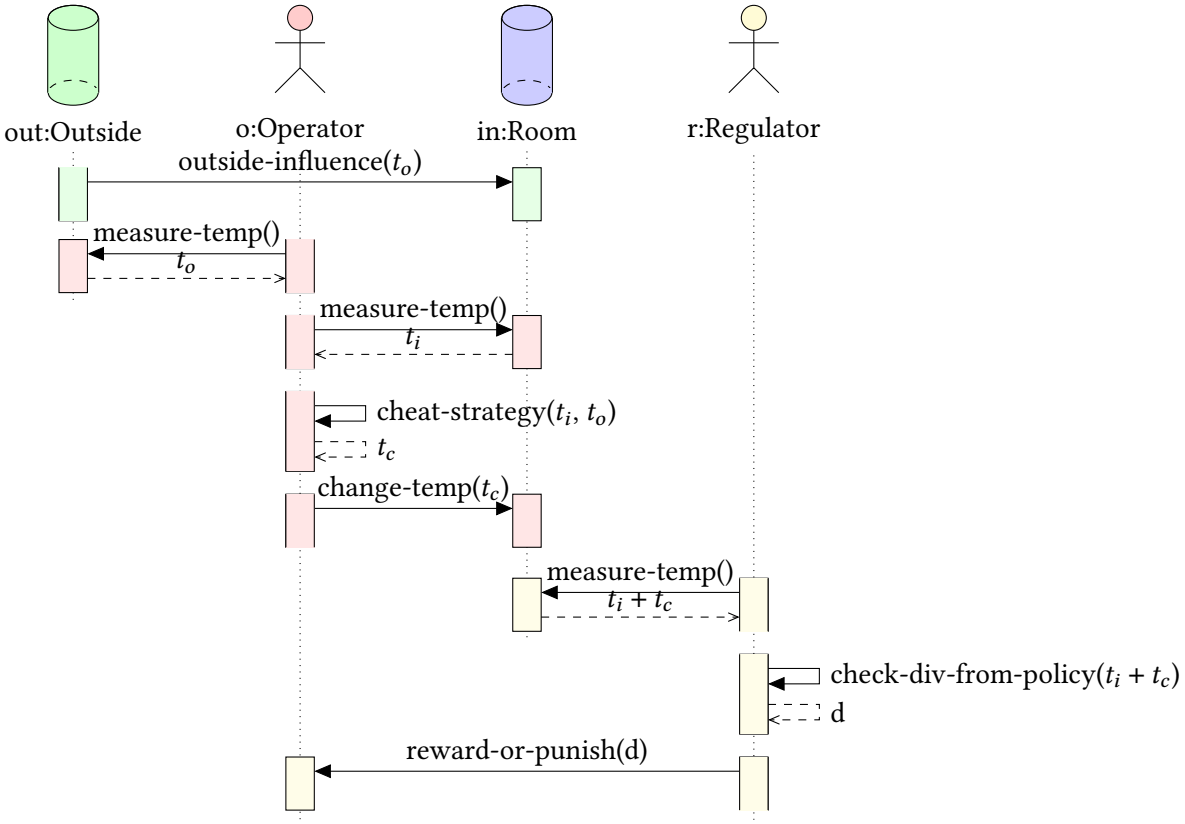


**Figure 3.** Sequence diagram of two-player thermostat model.

Here, the operator (red) and regulator (yellow) actions happen at different frequencies $f_o$ and $f_r$ with $f_r < f_o$ and $f_o$ = once per time step. Outside influence can be continuous in time, but no change faster than the maximal operator measurement frequency is relevant for our model.

**Remark 7.** Assuming instantaneous temperature exchange between the outside and the room (e.g. the room has no insulation), the operator can omit either one of the temperature measurements.

Given the above model, with $T_c = \sum t_c$ over all timesteps, the payoff functions for the operator are:

$$p_s^G(\hat{S}) = (40000 - T_c) - w_s^G(G_n, \hat{S}) \tag{4}$$

$$p_s^B(\hat{S}) = (40000 - T_c) - w_s^B(G_n, B_n, \hat{S}) \tag{5}$$

For plots of $w_s^G$ and $w_s^B$, see subfigures 2.2 and 2.3 below.

## 4.2. Empirical Analysis

### 4.2.1. Simulation Data

To get an intuition for how our example game plays out, given the above model and basic incentives, we simulate experiments and perform empirical analysis on it. We run experiments with $D_c = B(10, p_i)$, $p_i \in \{0, 0.1, 0.2, ..., 0.9, 1\}$, with $S = 10000$, $R_b = 40000$, and interpolation replacing missing values with the mean of available data.

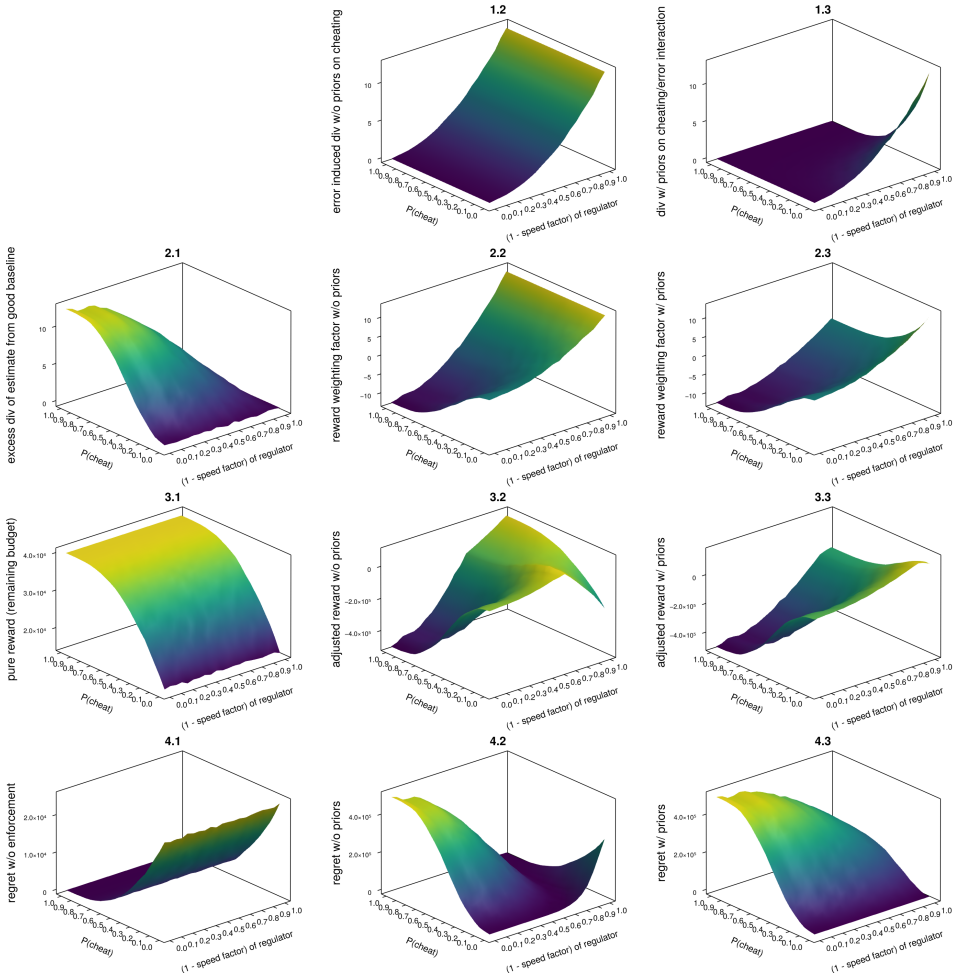


**Figure 4.** Experiments with $D_c = B(10, p_i)$, $p_i \in \{0, 0.1, 0.2, ..., 0.9, 1\}$.

The numbered subfigures from Figure 4 show the following:

**1.2** shows the pure error. (Error for $p = 0$ repeated along $p_i$.)

**1.3** shows the cheating prior. Since we know the full signal, we can compute $d_s$ for every signal/estimate pair.

**2.1** shows excess divergence. We don't assume to know any signal apart from the baseline, only observed estimates.

**2.2** payoff weights $w_s^G$ derived from pure error.

**2.3** payoff weights $w_s^B$ derived from cheating priors.

**3.1** base reward for operator w/o payoff weights.

**3.2** reward weighted entrywise by 2.2. according to (1).

**3.3** reward weighted entrywise by 2.3. according to (2).

**4.1** regret w/o reward weighting.

**4.2** regret corresponding to 3.2.: in-policy behaviour is the dominant strategy up to roughly 0.65 speed factor.

**4.3** regret corresponding to 3.3.: in-policy behaviour is dominant strategy in all speed regimes.

#### 4.2.2. Discussion

Looking at 1.2 and 1.3 in Figure 4, we can see that knowing how the conflation of the cheating strategy with the pure error gives us tighter divergence information compared to only the pure error, especially in the regimes with both high speed difference and high cheating probability.

Subtracting excess divergence (2.1) from either of the above gives us different weighting surfaces for the reward, the result of which is shown in 3.2 and 3.3, respectively, with 4.2 and 4.3 being the corresponding regret formulations.

We can see negative reward payments (i.e. punishment) in the low-speed difference and high cheat regimes in both cases, causing high regret to the operator. The weights derived from the pure error result in rewards for cheating in the high speed difference regime, though, i.e. the policy is too lenient.

The policy derived from the conflation (1.3) is tight enough in all speed regimes to incentivize in-policy behaviour, as regret reliably increases together with cheating everywhere.

**Remark 8.** The policy is encoded in the reward weighting surfaces. Assuming the regulator knows their measurement error and the observed excess divergence of the estimate from a good baseline signal, they can read off the weight they should apply to the base reward.

## 5. Future directions

We see this work as an initial step in exploring the problem space. Now, that we have a better idea about the model and the subgames being played, further research questions arise and opportunities for application are clearer.

### Composition of slow games

On the Anoma network, operators and regulators will often be internally coordinated in setups that can be modelled as a slow game. How do these games compose?

### Decomposition of a regulator

One example for the above mentioned internal coordination: Users of a specific service, each of which has a partial view of the outcomes produced by that service. How can they coordinate amongst each other, what are the incentive problems and how does that influence approximation bounds?

### Operator collusion in slow games

One of the goals of this line of research is to build a framework which enables regulators to create equilibria in which service providers compete on policy adherence to maximize regulator welfare. What if the operators collude? Can this be detected, or do conditions exist under which we can bound its influence?

### Applied modeling of Anoma

Two direct applications are mentioned in the examples section of the introduction: controller selection 1.2.1 and solver selection 1.2.2 in Anoma.

### Empirical pipeline

Once the Anoma network becomes operational, we hope to incorporate this framework into an empirical pipeline that can help inform decision making for users in practice.

### Transformation into a prior-free mechanism

By moving from a setting in which we assume a common prior over the underlying state, to the setting described in [CHJ20], where we use a learner to forecast the underlying state, it might be possible to transform our approach into a prior-free online mechanism, making its application feasible in practical settings.

For example, on the Anoma network: Users (which compose to the regulator) want transactions (TXs) to be ordered and included in blocks published by a controller[7] (which act as service providers). Users and controller agree

[7] See [Isa24]

on a specific mechanism, i.e. which metric is to be maximised by ordering of TXs in the blocks.

The controller is the only party that has exact knowledge about the order in which it receives the TXs, as well as the ability to potentially drop or include TXs, from which it builds a block. This means the controller has an informational advantage (knowing which blocks could be built) over the users, which it can use to defect from the agreed upon mechanism without it being detectable by the users, who can only (partially) observe outcomes.

Theorem 3 from [CHJ20], giving an upper bound for the principals (or here the users) expected regret in their setting, should provide an "upper" upper bound for extractable value in this example as well, with some caveats: Extractable value should only be a part of principal regret, the rest being agent regret, friction in different places etc.

Regarding the component terms of the theorem, we can say the following: The *cost of $\bar{\epsilon}$-informational robustness* depends how well we can solve for efficient, $\epsilon$-robust mechanisms for this specific use-case. Also, the state is revealed to the agent before it chooses an action, resulting in a larger informational advantage. *Agent regret* will depend on the complexity of the utility functions depending on TX ordering and available computational resources. For *forecast miscalibration*, we need to make a refinement: The state is only partially revealed to the principal, depending on how much data they can aggregate regarding the input TXs and how well the outcome can be estimated. The *discretization error* should be 0, since our problem is a discrete knapsack problem.

## 6. Acknowledgements

We thank the reviewers of the Agentic Markets Workshop at ICML 2024[8] for their helpful feedback on improving the exposition and refining this work.

[8] http://agenticmarkets.xyz/

# References


BEKS17. Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017. URL: https://doi.org/10.1137/141000671.

BM07. Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8(47):1307–1324, 2007. URL: http://jmlr.org/papers/v8/blum07a.html. (cit. on pp. 5 and 10.)

BM19. Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff, 2019. URL: https://arxiv.org/abs/1901.07821, arXiv:1901.07821. (cit. on pp. 5 and 7.)

CHJ20. Modibo Camara, Jason Hartline, and Aleck Johnsen. Mechanisms for a no-regret agent: Beyond the common prior, 2020. URL: https://arxiv.org/abs/2009.05518, arXiv:2009.05518. (cit. on pp. 6, 16, and 17.)

CKPD24. Tarun Chitra, Kshitij Kulkarni, Mallesh Pai, and Theo Diamandis. An analysis of intent-based markets, 2024. URL: https://arxiv.org/abs/2403.02525, arXiv:2403.02525. (cit. on p. 7.)

CRS23. Natalie Collina, Aaron Roth, and Han Shao. Efficient prior-free mechanisms for no-regret agents, 2023. URL: https://arxiv.org/abs/2311.07754, arXiv:2311.07754. (cit. on p. 7.)

GW11. Alexei A Gaivoronski and Adrian Werner. Stochastic programming perspective on the agency problems under uncertainty. In *Managing Safety of Heterogeneous Systems: Decisions under Uncertainties and Risks*, pages 137–167. Springer, 2011. (cit. on p. 6.)

Han57. James Hannan. Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957. (cit. on p. 5.)

Isa24. Sheff Isaac. Cross-Chain Integrity with Controller Labels and Endorsement. *Anoma Research Topics*, Jun 2024. URL: https://doi.org/10.5281/zenodo.10498996, doi:10.5281/zenodo.10498997. (cit. on p. 16.)

Ros73. Stephen A. Ross. The economic theory of agency: The principal's problem. *The American Economic Review*, 63(2):134–139, 1973. URL: http://www.jstor.org/stable/1817064. (cit. on p. 5.)

S+59. Claude E Shannon et al. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec*, 4(142-163):1, 1959. (cit. on p. 5.)

## A. Variation of cheating prior

We show another experiment, with different parameters for the cheating distribution: $D_c = B(3, p_i), p_i$ as above.

**1.2** shows how the pure error stays the same.

**1.3** shows how the conflation of pure error and cheating prior having a different shape.

**2.1** since in the excess divergence measurements, we only observe the estimates, this shape also changes.

**2.2-4.3** are analogous to Fig. 4, but derived from 1.3 and 2.1.

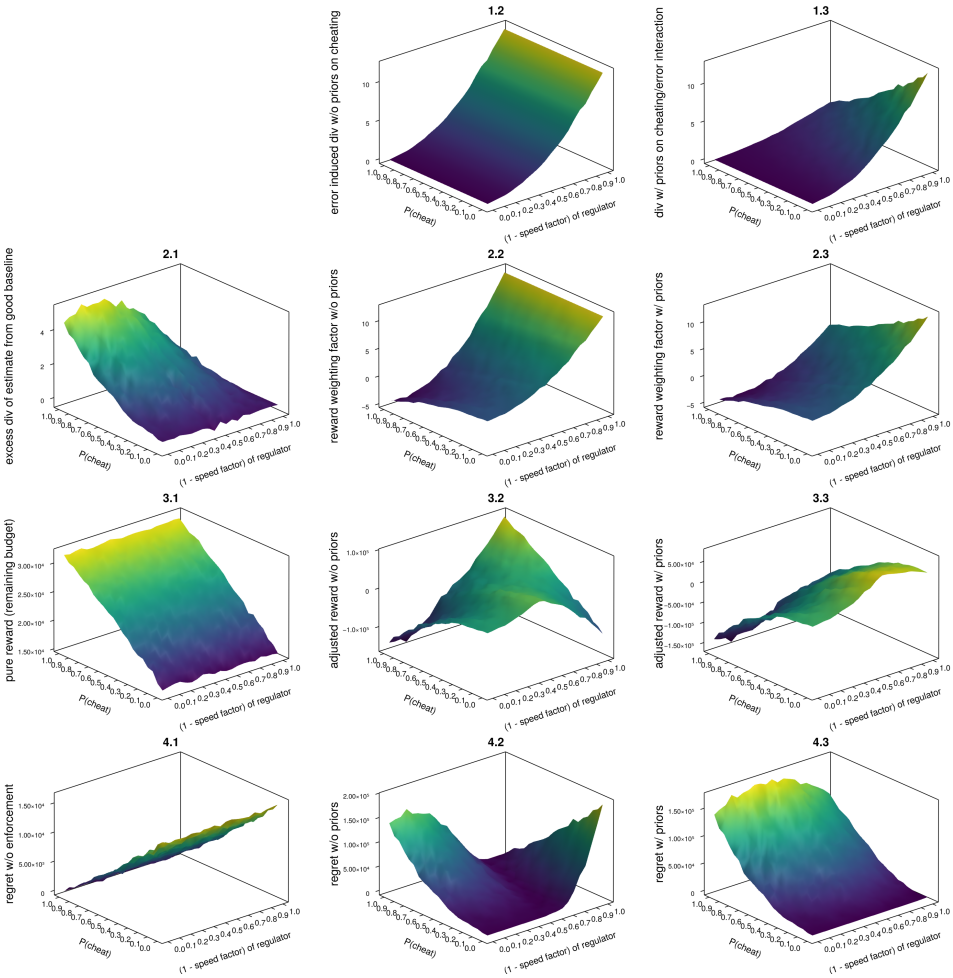


**Figure 5.** Experiments with $D_c = B(3, p_i), p_i$ as above, to show cheating prior where pure error is conflated with a different distribution.