SPAR Final Report

Market-Based Compute Permits for Frontier AI Safety

Kumari Neha Priya
kumari4@illinois.edu

Executive Summary

This paper examines market-based compute permits as a governance mechanism for frontier AI training. The core problem is that a compute permit regime may appear strict on paper while failing to constrain real compute use if its timing and enforcement rules are poorly designed. The analysis uses two stylized simulation tracks. The first studies banking rules under improving compute efficiency. It shows that unrestricted banking produces the largest late-period concentration of compute, while decay rules and banking caps reduce this effect only partially. The second simulation introduces a two-lab regulator model with an aggregate compute cap, monitoring probability, and fines for detected excess compute. This model shows that compliance depends on a simple threshold condition: the expected penalty for evasion must be at least as large as the permit price. Once permit scarcity pushes the price above that threshold, tighter caps reduce legal permit holdings, but actual compute remains almost flat as evasion rises.

The main implication is that compute permit design is not only about setting an aggregate cap. It is also about ensuring that banking rules, monitoring, and penalties keep actual compute responsive to policy. The paper recommends that regulators avoid unrestricted banking as a default, constrain any banking that is allowed, and design cap stringency and enforcement capacity jointly rather than separately.

1. Introduction & Problem Statement

Frontier AI training relies on large-scale compute infrastructure, which makes compute a plausible target for governance. Relative to other important inputs such as data and algorithms, compute is unusually measurable, concentrated, and tied to the development of frontier systems, which makes it a potentially useful intervention point for governance. A market-based compute permit system would aim to regulate access to high-stakes training compute while preserving some flexibility in how that compute is allocated over time.

A central design problem, however, is that permit rules may affect not only the total volume of compute used but also its timing. This matters if later periods are more capability-relevant because hardware and algorithms improve over time. Monitoring and verification are also relevant to the broader compute governance agenda, since proposals in this area often depend on whether large-scale training activity can be credibly detected or audited.

This paper focuses on two linked questions within compute permit design. First, can banking provisions unintentionally concentrate frontier training into a small number of later runs? Second, when does a tighter aggregate cap stop constraining actual compute because enforcement does not scale with permit scarcity? These questions are closely connected. A permit regime that allows too much temporal flexibility may shift compute into later, more capability-intensive periods. A permit regime that tightens caps without credible enforcement may reduce legal permit holdings while leaving actual compute use largely unchanged.

The relevant actors are frontier AI developers, cloud and compute providers, national regulators, standards bodies, and international institutions that may eventually coordinate compute-based oversight. The institutional setting is still emerging, so the goal here is not to evaluate an existing statutory regime. The goal is to clarify design conditions under which a market-based compute permit system would remain meaningful in practice.

2. Background & Context

Market-based permit systems are often justified on the grounds that they can achieve aggregate limits while preserving flexibility in how regulated actors respond. Stavins (2008) argues that, in the climate context, a well-designed cap-and-trade system can provide a credible aggregate constraint, create a price signal for regulated activity, and lower compliance costs by allowing

trading across firms. The paper also notes that design choices such as banking and borrowing may help manage cost uncertainty and smooth adjustment over time.

In the AI context, compute has increasingly been treated as a promising governance lever because it is detectable, excludable, quantifiable, and produced through a highly concentrated supply chain. Sastry et al. (2024) argue that these characteristics make compute especially relevant for regulatory visibility, resource allocation, and enforcement in frontier AI governance. Shavit (2023) examines one possible mechanism for verifying rules on large-scale neural network training through compute monitoring. The paper's relevance here is not that it resolves the full design of compute governance, but that it shows why monitoring and verification questions are central to any regime that aims to govern frontier training activity.

The present paper applies the logic of market-based permits to frontier AI compute rather than emissions. The specific focus is whether banking rules, while attractive as a flexibility mechanism, may also change the timing of compute use in ways that become governance-relevant when later-period training is more capability-intensive.

The carbon-market analogy is useful but incomplete. In emissions markets, banking can help firms smooth compliance costs across time. In compute governance, the same design feature may have a different effect because the capability-relevance of compute can change quickly. A unit of compute saved for later may enable a more powerful training run if hardware and model efficiency improve. This makes timing rules more central than they would be in a standard emissions setting.

The same concern applies to enforcement. A cap-and-trade system relies on the idea that regulated actors face a meaningful cost for exceeding their permitted activity. In a compute permit regime, that cost depends on detection probability, penalty size, and the market price of legal permits. If legal permits become scarce and expensive while enforcement remains weak, evasion may become attractive. The analysis below therefore treats banking and enforcement as two design layers of the same governance problem, keeping actual compute use responsive to policy rather than shifting it into delayed concentration or noncompliance.

3. Analysis

3.1 Analytical Framework

The analysis proceeds in two stages. The first stage examines banking rules under improving compute efficiency. The second stage examines the interaction between cap stringency and enforcement in a two-lab regulator model.

The first simulation uses a simple setup with one representative lab, a finite time horizon, fixed permit allocations per period, and improving compute efficiency over time. The second simulation uses two risk-neutral labs and one regulator. The regulator sets an aggregate compute cap, a monitoring probability, and a fine per unit of detected excess compute. Labs choose permit holdings and actual compute deployment. Evasion occurs when actual compute exceeds legal permit holdings.

The simulations are intentionally stylized. They are not forecasts of real-world compute markets. Their purpose is to isolate mechanisms that matter for institutional design: timing effects from banking, compliance thresholds under enforcement, and the point at which tighter caps stop reducing actual compute.

3.2 Banking Rules and Late-Period Concentration

The first analysis isolates one design question within compute permit systems: how banking rules affect the timing of frontier training under improving compute efficiency. The first-pass scenarios compare no banking, unlimited banking, banking with decay, and banking caps while holding other features constant and excluding imperfect enforcement.

As shown in Table 1, the no-banking regime produces the smallest peak run and the lowest concentration ratio, while unlimited banking produces the largest peak run and the highest concentration ratio. Banking with decay and banking caps reduce concentration relative to unlimited banking but not enough to bring outcomes close to the no-banking baseline.

Table 1: Baseline scenario comparison and concentration outcomes

Scenario	Peak run size	Concentration ratio
No banking	190.00	0.339
Unlimited banking	269.23	0.453
Banking with decay	253.96	0.447
Banking cap	247.00	0.421

Here, the concentration ratio is defined as the largest period's effective compute divided by total effective compute across all periods.

The no-banking case is the baseline for the banking analysis. It produces the lowest peak run size and the lowest concentration ratio because unused permits expire each period and cannot be saved for later. This keeps compute use more evenly distributed over time instead of shifting it into future periods. In policy terms, no banking is the strongest of the four simple designs for avoiding large, concentrated frontier runs, although it also gives labs the least flexibility.

Unlimited banking produces the largest peak run and the highest concentration ratio. Because compute becomes more effective in later periods, unrestricted carry-forward gives the lab an incentive to delay some permit use and accumulate a larger stock for the future. The result is a large final-period run rather than smoother compute use across time. A concentration ratio of 0.453 means that roughly 45 percent of total effective compute occurs in the single largest period, which is substantially higher than in the no-banking baseline. This result matters because even if total permits are unchanged, the timing of their use can still affect safety-relevant concentration.

Banking with decay reduces the buildup incentive, but only modestly in the baseline parameterization. Compared with unlimited banking, the peak run size is lower and the concentration ratio falls slightly. However, both measures remain much closer to the unlimited banking case than to the no-banking baseline. In the current setup, the decay parameter is 0.85, meaning that 85 percent of unused permits are carried forward into the next period. This weakens stockpiling incentives, but it does not prevent substantial late-period concentration.

The banking-cap scenario performs better than unlimited banking and somewhat better than the decay scenario in terms of concentration. The cap limits how much banked permits can be used in a single period, which prevents the lab from turning all accumulated stock into one extremely large run. The final-period spike is still present, but it is smaller than under unlimited banking. In the current implementation, this is a cap on the use of banked permits per period, not a direct cap on the total banked stock. The relatively flat banked-permit path in early periods reflects the model's behavioral rule: the lab withdraws roughly 30 banked permits in a period and then replenishes the bank with newly unused current-period permits. The flat path is therefore a consequence of the interaction between the usage cap and the behavioral rule, not a direct stock cap.

Figure 1: Effective compute by period across banking scenarios

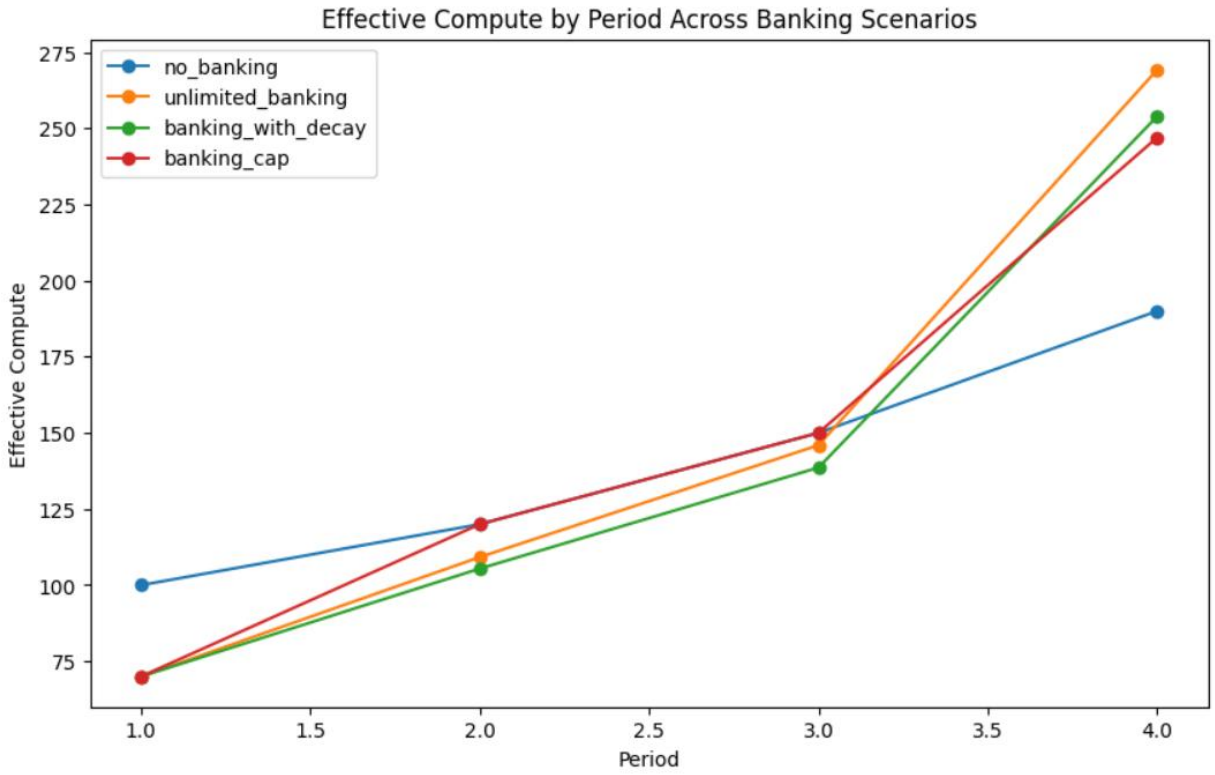


Figure 1 shows the timing mechanism behind these results. In the no banking case, effective compute rises more smoothly because permits must be used or lost each period. In the banking scenarios, compute use shifts toward later periods, especially under unlimited banking. Since later-period compute is assumed to be more effective, this shift creates larger effective runs. The especially large final-period spike should also be read partly as a finite-horizon effect: once the model reaches the last period, all remaining saved permits are used because there is no later period to bank into.

Figure 2: Banked permits over time across banking scenarios

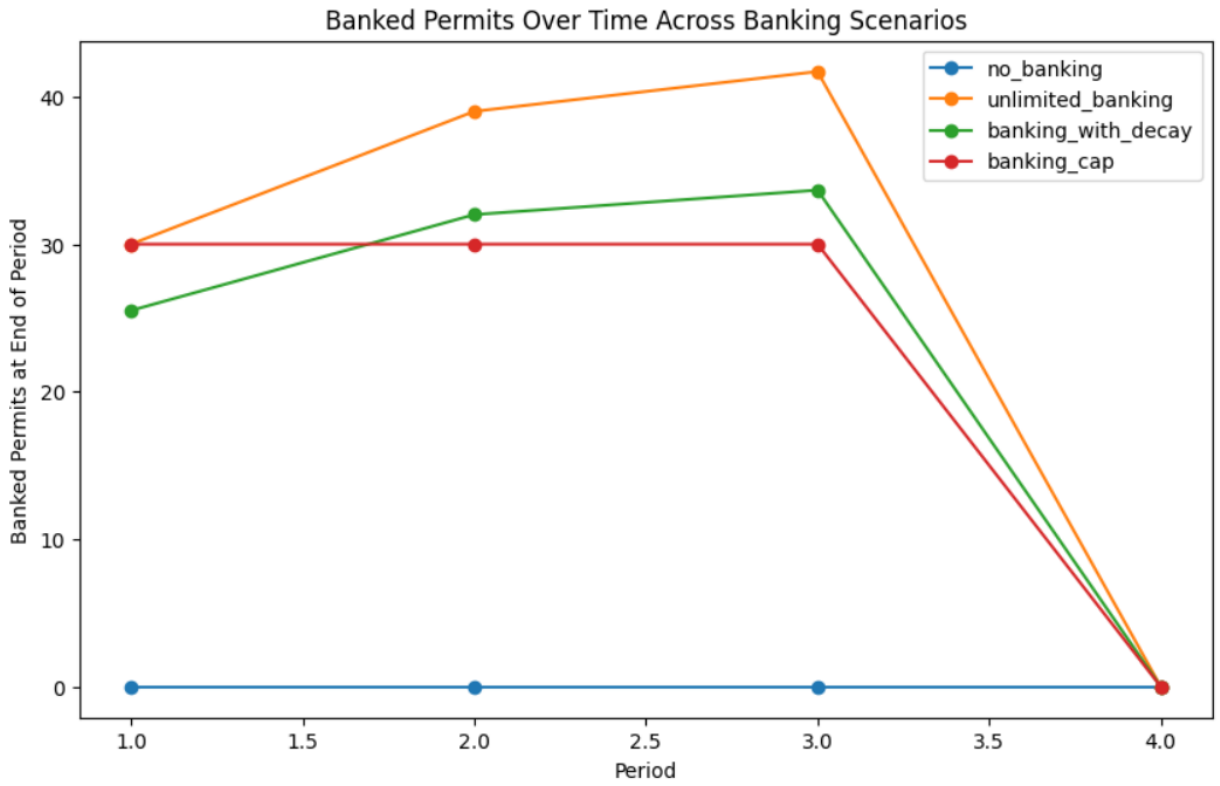


Figure 2 explains the mechanism behind Figure 1. Under no banking, banked permits remain at zero because unused permits expire immediately. Under unlimited banking, permits accumulate the most, which is why that scenario produces the largest late-period spike in effective compute. The decay scenario also allows accumulation, but less of it, since some stored permits lose value over time. The banking-cap scenario keeps banked permits flatter by restricting how much of the bank can be deployed in a given period. Taken together, Figures 1 and 2 show that differences in peak run size are driven by differences in how much permit stock can build up before being used.

The sensitivity checks add two further insights. Rows 0-2 in Table 2 report the decay sensitivity results, and rows 3-5 report the banking-cap sensitivity results.

Table 2: Sensitivity checks for decay and banking caps

Parameter value	Peak run size	Concentration ratio	Scenario type
0.95	263.981084	0.450961	decay
0.85	253.955211	0.447122	decay
0.70	240.038590	0.441768	decay
0.50	247.000000	0.420784	cap

0.30	247.000000	0.420784	cap
0.15	218.500000	0.421815	cap

As shown in Table 2 and Figure 3, stronger decay reduces late-period buildup in the expected direction. As the carry-forward rate falls from 0.95 to 0.70, both peak run size and concentration ratio decline, suggesting that decay can moderate temporal concentration, although only gradually in the current setup.

Figure 3: Peak run size under different decay rates

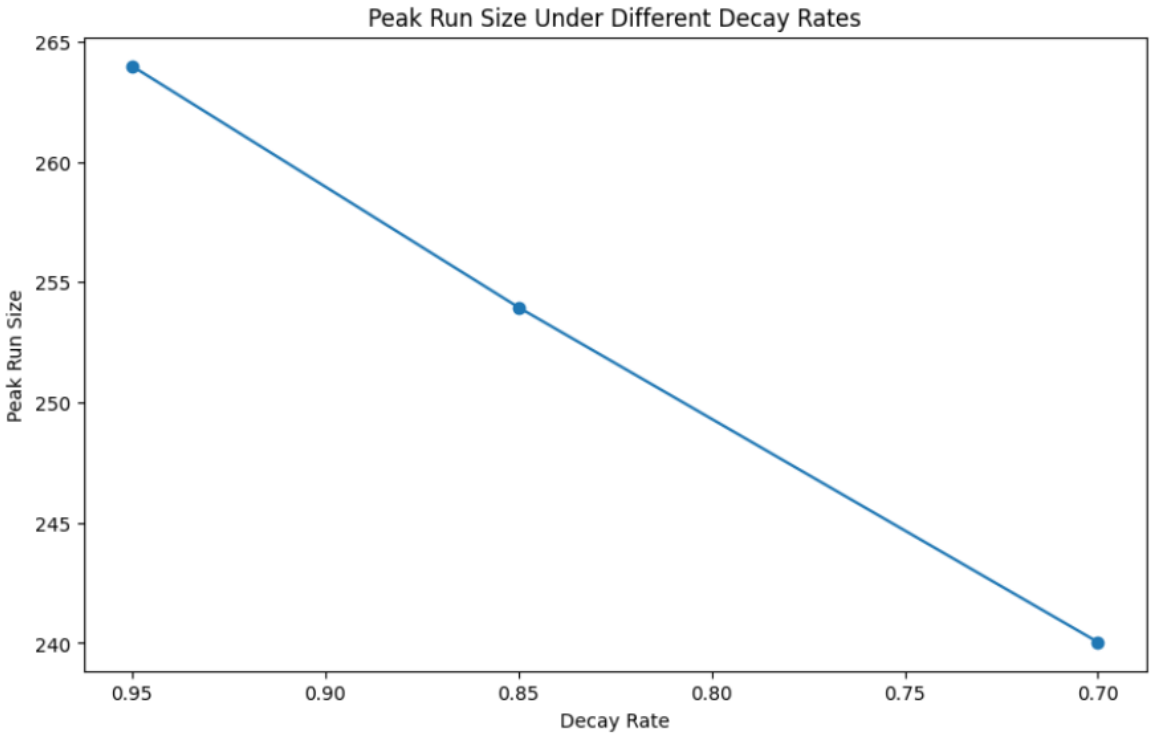
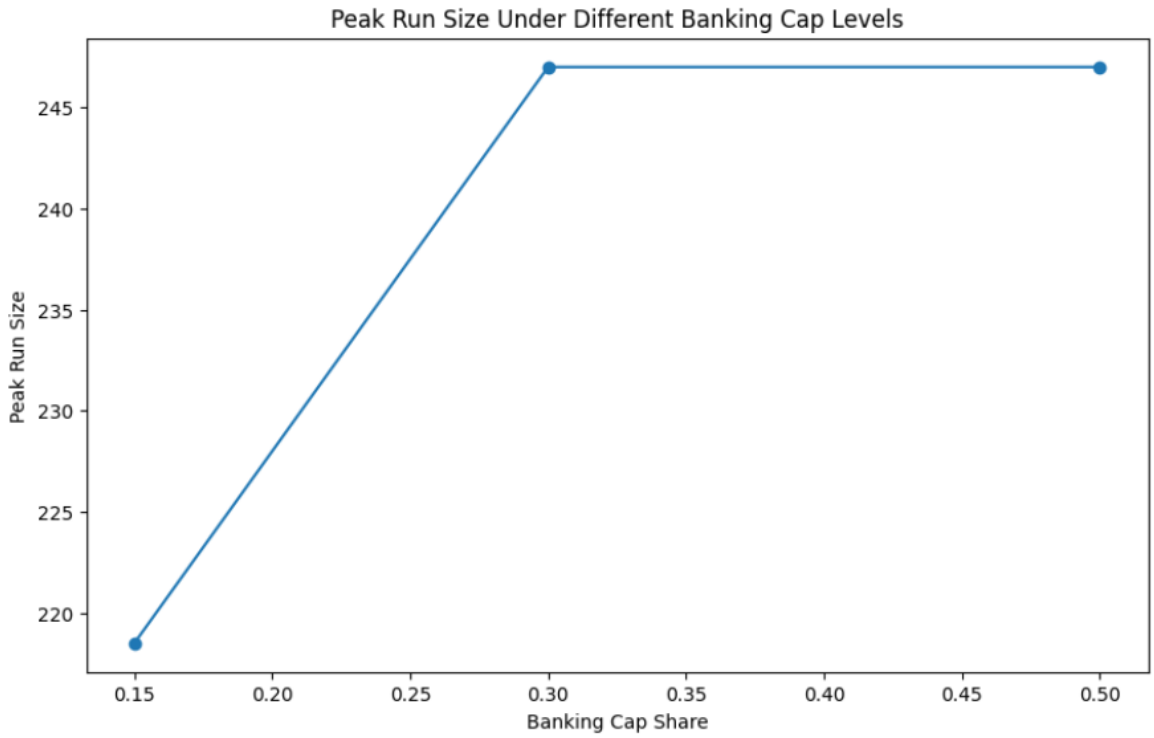


Table 2 and Figure 4 also show that the banking-cap results follow more of a threshold pattern than a smooth response. Cap shares of 0.50 and 0.30 produce identical outcomes, implying that neither constraint binds meaningfully under the baseline behavioral rule. A tighter cap of 0.15 reduces peak run size substantially, but has little effect on the concentration ratio, indicating that stricter caps may reduce absolute spike size without greatly changing the overall timing pattern of compute use.

Figure 4: Peak run size under different banking cap levels



Taken together, the banking simulations support the core intuition behind the banking puzzle. When compute becomes more effective over time, allowing permits to be carried forward creates an incentive to delay some use and accumulate for the future. This leads to larger and more concentrated frontier runs in later periods, even when total permit availability is unchanged. Unlimited banking produces the strongest version of this effect, while decay and banking caps reduce it somewhat but do not eliminate it. The policy challenge is therefore not only how many permits are issued but also how permit rules shape the timing of compute use across periods.

3.3 Enforcement Model

The second analysis extends the paper from timing rules to enforcement design. It uses a two-lab regulator model to study when a compute permit regime remains binding in practice. The model has two risk-neutral labs and one regulator. Each lab chooses compute deployment to maximize expected profit from frontier AI training. The regulator chooses policy parameters to limit socially costly concentrations of frontier compute while preserving beneficial training activity.

The regulator sets three policy variables: an aggregate compute cap, denoted by $\bar{A}$, a monitoring probability, denoted by μ, and a fine per unit of detected excess compute, denoted

by F. Each lab chooses permit holdings $h_i \geq 0$ and actual compute deployment $c_i \geq 0$. Evasion occurs when a lab's actual compute exceeds its permit holdings, $c_i > h_i$.

The model proceeds in three stages. First, the regulator sets $(\bar{A}, \mu, F)$. Second, a competitive permit market clears at price τ, subject to the aggregate cap $\bar{A}$. Labs then choose permit holdings and compute deployment while taking the permit price as given. Third, each lab is independently audited with probability μ. If excess compute is detected, the lab pays a fine F per unit of evasion.

Lab i's payoff is:

$$\Pi_i = R_i(c_i) - \tau h_i - \mu F \max(c_i - h_i, 0)$$

where $R_i(c_i)$ is a concave training-value function, τh_i is the cost of holding permits, and $\mu F \max(c_i - h_i, 0)$ is the expected penalty for evasion. Social welfare is represented as:

$$W = \sum_i R_i(c_i) - S(c_1 + c_2)$$

where $S(\cdot)$ is a convex safety-cost function. The model assumes observable and similar training-return functions rather than private information. This keeps the enforcement mechanism clean and easier to connect to the simulation.

The key compliance condition follows from a marginal comparison. A lab deciding whether to cover one additional unit of compute legally compares the cost of buying one more permit with the expected cost of evading. The marginal cost of compliance is the permit price, τ. The marginal expected cost of evasion is μF. Compliance is therefore sustained when:

$$\boldsymbol{\mu F \geq \tau}$$

If this condition fails, evasion becomes attractive on the margin. A tighter cap raises permit scarcity, which increases the equilibrium permit price τ. If monitoring and fines do not rise with scarcity, the expected penalty remains fixed while the cost of legal compliance increases. This produces the tightening-evasion spiral: tighter caps can increase evasion rather than effective compliance, making the permit regime less meaningful in practice.

3.4 Cap Tightening and the Compliance Threshold

The enforcement simulation uses a baseline monitoring probability of $\mu = 0.4$ and a fine of $F = 80$, so the expected penalty is fixed at $\mu F = 32$. This parameterization creates a clear test of the compliance threshold: caps remain effective while the permit price stays below 32, and evasion becomes attractive once the permit price rises above 32.

Table 3: Cap tightening, permit prices, and evasion

Aggregate cap $\bar{A}$	**Permit price τ**	**Expected penalty μF**	**Total permit holdings**	**Total actual compute**	**Total evasion**
220	20.0	32.0	220.00	220.00	0.00
200	30.0	32.0	200.00	200.00	0.00
180	40.0	32.0	180.00	196.00	16.00
160	50.0	32.0	160.00	196.00	36.00
140	60.0	32.0	140.00	196.00	56.00
120	70.0	32.0	120.00	196.00	76.00
100	80.0	32.0	100.00	196.00	96.00

Table 3 shows the same threshold logic in the simulation outputs. At $\bar{A} = 220$ and $\bar{A} = 200$, the permit price remains below the expected penalty, so the compliance condition holds and total evasion is zero. At $\bar{A} = 180$, the permit price rises to $\tau = 40$, which exceeds $\mu F = 32$. The compliance condition fails. At this cap level, Lab 1 holds 100 permits but deploys 108 units of compute, while Lab 2 holds 80 permits but deploys 88 units. Total legal permit holdings equal 180, but total actual compute rises to 196, producing 16 units of evasion.

Figure 5: Permit price vs. expected penalty

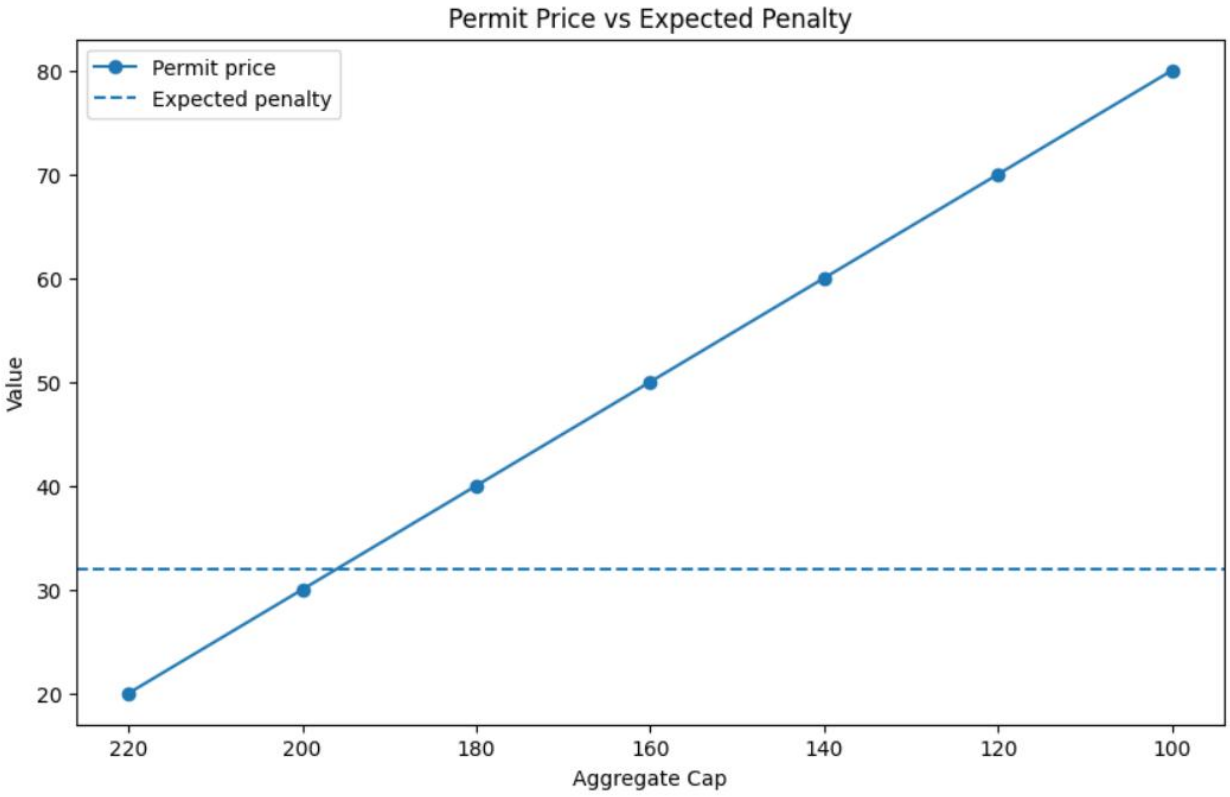


Figure 5 shows the central enforcement result. As the aggregate cap tightens, permit scarcity raises the equilibrium permit price. The expected penalty remains fixed at 32 in the baseline parameterization. Compliance holds while the permit price remains below that threshold. Once the cap tightens enough for the permit price to exceed the expected penalty, evasion becomes attractive. This crossing point marks the shift from a binding permit regime to one in which enforcement no longer keeps pace with scarcity.

Figure 6: Total evasion under different cap levels

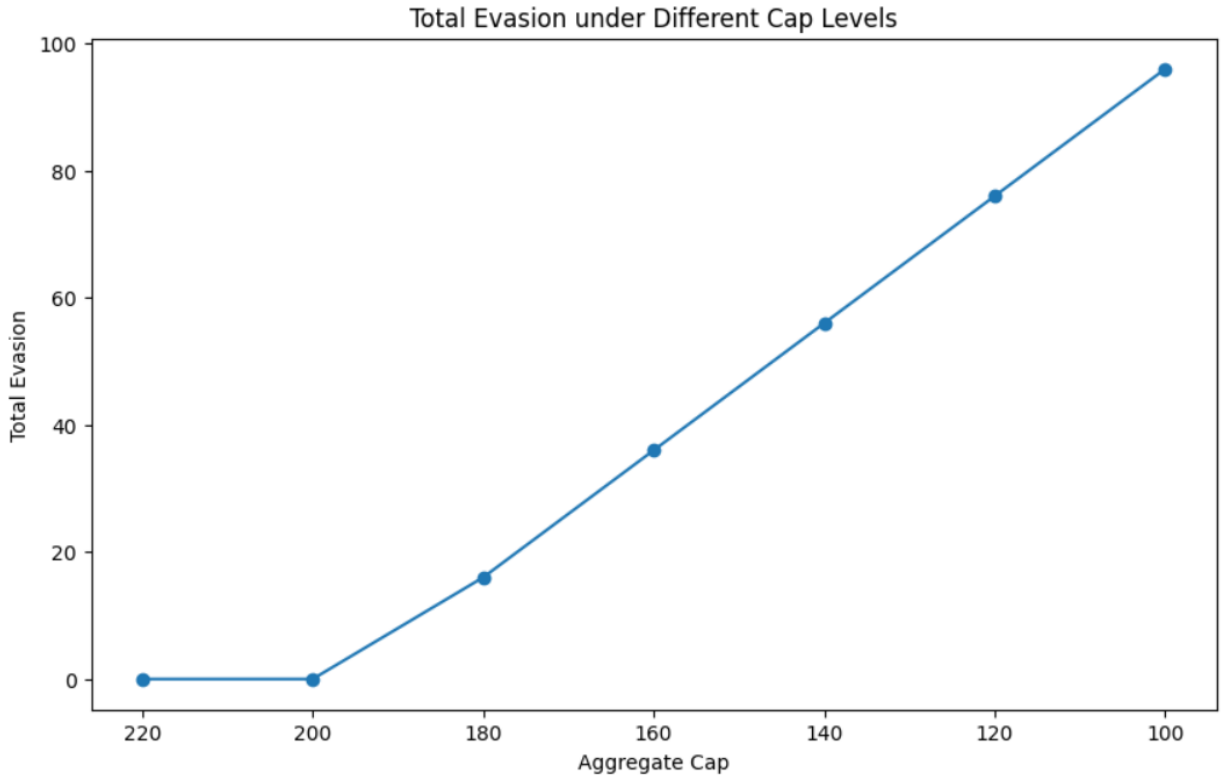


Figure 6 shows the behavioral consequence of crossing the compliance threshold. Evasion is zero at looser cap levels because the expected penalty exceeds the permit price. Once the permit price rises above the expected penalty, evasion begins and increases sharply as the cap tightens further. This result shows that tighter caps can increase noncompliance if enforcement does not scale with scarcity.

Figure 7: Actual compute vs. legal permit holdings

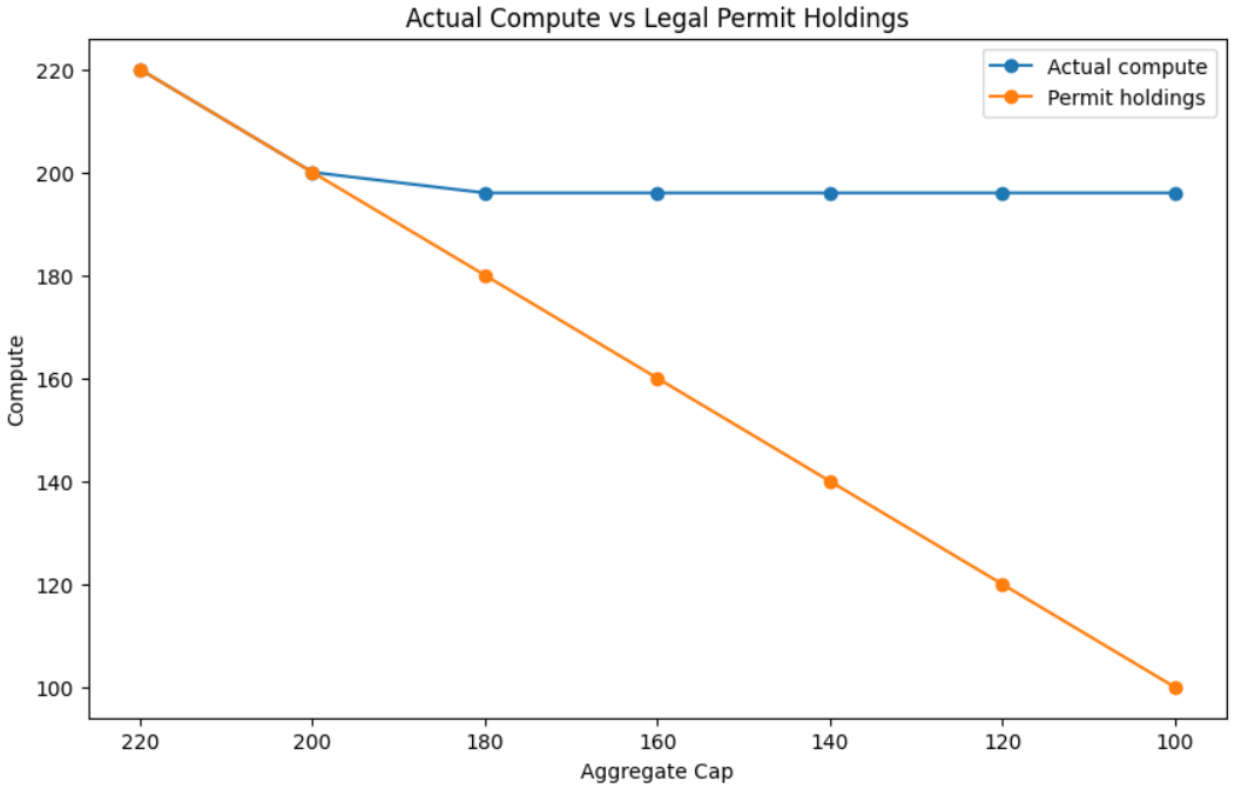


Figure 7 is the clearest policy result. Before the threshold is crossed, actual compute and legal permit holdings move together. After the threshold is crossed, legal permit holdings continue to fall as the cap tightens, but actual compute remains almost flat. The difference between the two is filled by evasion. This means that beyond the compliance threshold, tighter caps reduce legal compliance more than they reduce real compute use.

This is the central result of the paper. Tighter caps are effective only while enforcement remains strong enough to make legal compliance at least as attractive as evasion. Once permit scarcity pushes the permit price above the expected penalty threshold, the regime becomes less constraining in practice. Legal permit holdings fall, but actual compute remains high because the difference is filled by evasion.

3.5 Fines, Monitoring, and Counterproductive Caps

The policy grid extends the enforcement simulation by varying fines and monitoring. This directly answers three design questions: when raising fines is enough, when monitoring must increase, and when tighter caps alone become counterproductive. Tables 4 and 5 report selected threshold rows from the fine and monitoring sensitivity checks; the full grids are reported in Appendix Tables A1 and A2.

Raising fines is sufficient when it pushes the expected penalty above the permit price. At $\bar{A} = 200$, compliance is restored once $F = 80$ because $\mu F = 32$ exceeds $\tau = 30$. At $\bar{A} = 180$, a higher fine is needed: compliance is restored only once $F = 100$ because $\mu F = 40$. At $\bar{A} = 140$, compliance is restored only at $F = 160$ because $\mu F = 64$ exceeds $\tau = 60$. This shows that fines can restore compliance at moderate scarcity, but the required fine rises sharply as the cap tightens.

Table 4: Fine sensitivity at selected cap levels

$\bar{A}$	μ	Fine F	τ	μF	Compliance	Total actual compute	Total evasion
200	0.4	40	30.0	16.0	False	228.00	28.00
200	0.4	60	30.0	24.0	False	212.00	12.00
200	0.4	80	30.0	32.0	True	200.00	0.00
180	0.4	80	40.0	32.0	False	196.00	16.00
180	0.4	100	40.0	40.0	True	180.00	0.00
140	0.4	120	60.0	48.0	False	164.00	24.00
140	0.4	160	60.0	64.0	True	140.00	0.00

Monitoring must increase when fixed fines are no longer enough to keep the expected penalty above the permit price. With F fixed at 80, monitoring of $\mu = 0.4$ is sufficient at $\bar{A} = 200$ because $\mu F = 32$ exceeds $\tau = 30$. At $\bar{A} = 180$, $\mu = 0.4$ is no longer enough, and monitoring must rise to at least $\mu = 0.6$. At $\bar{A} = 140$, even $\mu = 0.6$ is insufficient; monitoring must rise to at least $\mu = 0.8$. The implication is that static enforcement is not enough under tighter caps. If fines are legally or politically constrained, monitoring intensity must scale with scarcity.

Table 5: Monitoring sensitivity at selected cap levels

$\bar{A}$	μ	Fine F	τ	μF	Compliance	Total actual compute	Total evasion
200	0.1	80	30.0	8.0	False	244.00	44.00
200	0.2	80	30.0	16.0	False	228.00	28.00
200	0.4	80	30.0	32.0	True	200.00	0.00
180	0.4	80	40.0	32.0	False	196.00	16.00
180	0.6	80	40.0	48.0	True	180.00	0.00
140	0.4	80	60.0	32.0	False	196.00	56.00
140	0.6	80	60.0	48.0	False	164.00	24.00

140	0.8	80	60.0	64.0	True	140.00	0.00

Tighter caps alone become counterproductive once the permit price exceeds the expected penalty threshold and actual compute stops responding meaningfully to further tightening. In the simulation, that threshold is crossed between $\bar{A} = 200$ and $\bar{A} = 180$. From $\bar{A} = 180$ onward, tighter caps reduce legal permit holdings, but total actual compute remains almost flat while evasion rises. In policy terms, this means that scarcity without credible enforcement can weaken the regime. The cap becomes stricter formally but less effective in practice.

4. Recommendations

The results support four recommendations for compute permit design. First, unrestricted banking should not be treated as a default feature of an initial permit regime. As shown in Table 1, unlimited banking produces the largest late-period spike and the greatest temporal concentration of compute under improving efficiency. A simpler regime with tighter temporal limits is likely to be easier to interpret, administer, and revise.

If banking is allowed, it should be introduced only with explicit constraints. The sensitivity checks in Table 2 indicate that stronger decay and tighter caps on the use of banked permits can reduce late-period buildup, although the size of the effect depends on parameterization. The policy implication is not that one rule has already been validated, but that flexibility should be introduced cautiously and only in forms that meaningfully limit concentrated frontier runs. Regulators designing pilot compute permit regimes should therefore specify whether permits expire, decay, or can be banked only within bounded limits.

Second, cap stringency and enforcement capacity should be designed jointly. The enforcement simulation shows that a tighter cap remains meaningful only while the expected penalty for evasion remains at least as large as the permit price. If the cap tightens while monitoring and penalties remain fixed, the permit price can rise above the expected penalty threshold. At that point, the regime reduces legal permit holdings but does not reduce actual compute proportionally. Regulators should therefore treat the aggregate cap, monitoring probability, and fine schedule as linked policy variables, not separate design choices.

Third, fines and monitoring should be used as complementary enforcement tools. Raising fines can restore compliance when the required fine is legally and politically feasible. The fine sensitivity results show that higher fines can bring expected penalties back above permit prices at moderate cap levels. However, as caps become tighter, the required fines rise sharply. If fines face legal, political, or credibility constraints, monitoring must increase instead. This means that

regulators and audit institutions should plan for enforcement capacity to scale with permit scarcity. A static monitoring regime may be adequate under loose caps but inadequate under tight caps.

Fourth, regulators should be cautious about tightening caps without first checking whether enforcement is strong enough to preserve compliance. The simulations show that tighter caps can backfire once scarcity outpaces enforcement. In that region, legal holdings fall while actual compute remains high and evasion rises. This does not mean that tight caps are undesirable. It means that cap tightening should be conditional on credible monitoring and penalties. A compute permit regime that looks strict but is easy to evade may create false confidence and reduce the practical value of the policy.

5. Conclusion

The analysis shows that compute permit design is not only about how many permits exist. It is also about how permit rules shape timing, concentration, and compliance. In the banking simulation, unrestricted banking produces the strongest late-period buildup of compute, while decay and tighter caps moderate that effect to varying degrees. In the enforcement simulation, the core result is sharper: cap stringency and enforcement capacity must be designed jointly. Once the permit price exceeds the expected penalty for evasion, tighter caps reduce legal permit holdings, but actual compute remains high because evasion fills the gap.

The broader lesson is that a market-based compute permit regime should be evaluated by whether it constrains real compute use, not only by whether it imposes a formal cap. Banking rules determine whether compute shifts across time. Enforcement rules determine whether the cap remains binding in practice. A credible compute permit system therefore needs simple timing rules, explicit banking constraints, and enforcement capacity that scales with scarcity. Without those conditions, the regime risks becoming formally strict but substantively weak.

References

Sastry, G., et al. (2024). *Computing power and the governance of artificial intelligence*. arXiv. https://arxiv.org/abs/2402.08797

Shavit, Y. (2023). *What does it take to catch a Chinchilla? Verifying rules on large-scale neural network training via compute monitoring*. arXiv. https://arxiv.org/abs/2303.11341

Stavins, R. N. (2008). *A meaningful U.S. cap-and-trade system to address climate change*. *Harvard Environmental Law Review, 32*, 293–371.

Appendix

Table A1: Fine sensitivity results

Aggregate cap $\bar{A}$	Monitoring probability μ	Fine F	Permit price τ	Expected penalty μF	Compliance condition met	Total actual compute	Total evasion
200	0.4	40	30.0	16.0	False	228.00	28.00
200	0.4	60	30.0	24.0	False	212.00	12.00
200	0.4	80	30.0	32.0	True	200.00	0.00
200	0.4	100	30.0	40.0	True	200.00	0.00
200	0.4	120	30.0	48.0	True	200.00	0.00
200	0.4	160	30.0	64.0	True	200.00	0.00
180	0.4	40	40.0	16.0	False	228.00	48.00
180	0.4	60	40.0	24.0	False	212.00	32.00
180	0.4	80	40.0	32.0	False	196.00	16.00
180	0.4	100	40.0	40.0	True	180.00	0.00
180	0.4	120	40.0	48.0	True	180.00	0.00
180	0.4	160	40.0	64.0	True	180.00	0.00
140	0.4	40	60.0	16.0	False	228.00	88.00
140	0.4	60	60.0	24.0	False	212.00	72.00
140	0.4	80	60.0	32.0	False	196.00	56.00
140	0.4	100	60.0	40.0	False	180.00	40.00
140	0.4	120	60.0	48.0	False	164.00	24.00
140	0.4	160	60.0	64.0	True	140.00	0.00

Table A2: Monitoring sensitivity results

Aggregate cap $\bar{A}$	Monitoring probability μ	Fine F	Permit price τ	Expected penalty μF	Compliance condition met	Total actual compute	Total evasion
200	0.1	80	30.0	8.0	False	244.00	44.00
200	0.2	80	30.0	16.0	False	228.00	28.00
200	0.4	80	30.0	32.0	True	200.00	0.00
200	0.6	80	30.0	48.0	True	200.00	0.00
200	0.8	80	30.0	64.0	True	200.00	0.00

200	1.0	80	30.0	80.0	True	200.00	0.00
180	0.1	80	40.0	8.0	False	244.00	64.00
180	0.2	80	40.0	16.0	False	228.00	48.00
180	0.4	80	40.0	32.0	False	196.00	16.00
180	0.6	80	40.0	48.0	True	180.00	0.00
180	0.8	80	40.0	64.0	True	180.00	0.00
180	1.0	80	40.0	80.0	True	180.00	0.00
140	0.1	80	60.0	8.0	False	244.00	104.00
140	0.2	80	60.0	16.0	False	228.00	88.00
140	0.4	80	60.0	32.0	False	196.00	56.00
140	0.6	80	60.0	48.0	False	164.00	24.00
140	0.8	80	60.0	64.0	True	140.00	0.00
140	1.0	80	60.0	80.0	True	140.00	0.00