Strategic Interaction Among AI Agents: Evidence from Payment Systems*

Furkan Danisman[1], Ajit Desai[2], Janet Jiang[2], Francisco Rivadeneyra[2], and Talha Azad[2]

[1]University of Toronto & Vector Institute
[2]Bank of Canada

September 4, 2026

Abstract

We provide a framework for understanding the use of AI agents in interactive strategic financial systems, where the actions of one agent affect the incentives and opportunities of others. We illustrate the framework using an intraday liquidity management game in real-time gross settlement systems, which offers a controlled setting featuring scarce liquidity resources, endogenous strategic uncertainty, and interdependent payment flows. We conduct three main treatments that vary prompt design and equilibrium structure. With strategy-informed instructions and a unique equilibrium, reasoning capable AI agents learn to strategically delay payments, preserve liquidity for future obligations, and converge to equilibrium. Holding the economic environment fixed but removing explicit strategy explanations substantially reduces convergence. When the same basic game admits multiple equilibria, independent sessions converge to different parts of the equilibrium set. Structured reasoning texts about decision rationales help diagnose the strategic considerations that lead to different outcomes. Therefore, the behavior of interacting AI agents depends on both how the environment is represented to them and the strategic structure of the environment.

Keywords: AI agents; large language models; strategic interaction; payment systems; liquidity management; experimental economics

JEL Codes: A12, C7, D83, E42, E58

*For their comments, we thank the participants at the AI in Banking and Financial Services Workshop, CEBRA, the Canadian Economic Association Meeting, the International Conference on Payments and Securities Settlement, the Bank of Finland Seminar on Financial Market Infrastructures, and seminars at Bank of Canada and Simon Fraser Universality. The views expressed are solely those of the authors and do not necessarily reflect the views of the Bank of Canada.

1 Introduction

Artificial intelligence (AI) is increasingly moving from prediction and content generation toward autonomous decision making. In an agentic system, an AI model observes an environment, chooses actions, receives feedback, and adapts subsequent decisions. As organizations introduce such systems more broadly, multiple AI agents may make decisions that affect one another. An important question is therefore no longer only whether an AI agent can make a good decision in isolation, but whether interacting AI agents can recognize strategic interdependence, adapt to counterpart behavior, and reach stable outcomes through repeated interaction.

This question is particularly important in environments in which one agent's action changes the opportunities available to others. An agent may need to conserve resources for future contingencies, anticipate how a counterparty will respond, or take an action today in order to generate a favorable response tomorrow. Such considerations are familiar from game theory, but they pose a distinct challenge for autonomous AI systems. Pretrained large language models (LLMs) possess substantial general knowledge and reasoning capabilities, yet they are not typically trained for the particular strategic environments in which they may ultimately be introduced. Whether they can translate a natural-language description of an institution into strategically coherent behavior—and how that behavior depends on the information supplied to them—is therefore an empirical question.

We study these questions in the context of intraday liquidity management in real-time gross settlement (RTGS) payment systems. RTGS systems settle large and time-critical payments among financial institutions individually and irrevocably. This reduces settlement risk but requires participants to actively manage intraday liquidity. Participants can obtain liquidity in advance, but doing so is costly. Alternatively, they can delay outgoing payments and rely partly on incoming payments from other participants, at the cost of delayed settlement and potentially costly end-of-day borrowing. Because one participant's outgoing payment becomes another participant's incoming liquidity, payment decisions are strategically interdependent: a payment sent today can relax a counterparty's future liquidity constraint and thereby affect the liquidity that returns later.

We focus on a stylized two-agent cash-management game in which one participant faces an indivisible future payment. The indivisibility creates a nontrivial intertemporal strategic problem. The agent can use its liquidity to reduce current payment delays, but doing so may leave insufficient liquidity to execute a future lumpy payment. At the same time, the agent's current payment affects the liquidity available to its counterparty and hence potentially affects future incoming payments. Optimal behavior, therefore, requires the agent to balance current delay costs, future liquidity needs, borrowing costs, and the endogenous response of the other participant. Under our baseline parameterization, the game has a unique pure-strategy subgame-perfect Nash equilibrium in which the agent strategically delays part of its current payment and retains a liquidity buffer for the future indivisible obligation.

This environment provides a useful laboratory for studying strategic AI agents. The institutional rules, state variables, and payoff consequences can be stated precisely, and the underlying game can be solved analytically or numerically, giving us a theoretical benchmark against which agent behavior can be evaluated. Yet the equilibrium strategy is not mechanically implied by a one-period optimization problem. Agents must understand the dynamic link between current and future liquidity, account for the counterpart's behavior, and recognize when delaying a payment is strategically beneficial. The setting therefore allows us to study whether general-purpose LLMs can develop strategically coherent behavior through repeated interaction without task-specific model training.

We develop an experimental framework that connects reasoning-capable LLM agents to repeated interactive payment games implemented in `oTree`. Each LLM acts as a cash manager and receives information about its role, the rules of the payment system, its available actions, its objective function, the current state of the game, and the history of previous rounds. Agents choose initial liquidity and subsequent payment decisions, and each agent's outgoing payments become the other's incoming liquidity. The agents are pretrained general-purpose models rather than policies trained specifically for the game. They adapt through the contextual information and history supplied during repeated play. We also require agents to provide structured decision rationales alongside their choices. These records should not be interpreted as direct observations of the model's internal computational process, but they provide an auditable account of the considerations the agents report using, and allow us to examine how stated strategic considerations evolve with observed behavior.

Our first question is whether interacting LLM agents can discover the equilibrium strategy in this nontrivial dynamic game. In the strategy-informed version of the experiment, agents receive the formal rules and payoff structure together with explicit descriptions of economically relevant strategic relationships, including liquidity recycling and the possibility that retaining liquidity can facilitate a future indivisible payment. Starting from heterogeneous initial choices, the agents experiment with alternative liquidity and payment policies, form expectations about incoming payments from past outcomes, and eventually converge to the unique equilibrium. The decision records show a corresponding evolution in the considerations reported by the agents: they move from relatively simple comparisons of liquidity, delay, and borrowing costs toward explicit use of observed inflow patterns, the relationship between current payments and future incoming liquidity, and the value of preserving a buffer for the lumpy obligation.

Our second question is how this strategic behavior depends on the information provided to the agent. We therefore hold the economic game fixed and vary the prompt. In the strategy-informed treatment, the prompt describes both the formal rules and several strategically relevant relationships embedded in those rules. In the rules-only treatment, agents receive the same payoff parameters, institutional rules, state variables, objective function, and history of previous play, but the explicit strategic explanations are removed. This comparison separates an agent's ability to act strategically when the relevant mechanism is represented explicitly in its context from its ability to infer that mechanism from the formal environment and its own experience. The difference is substantial: under the rules-only treatment, four out of five sessions fail to converge to the theoretical equilibrium. The decision rationales also differ across treatments. Although agents in both treatments can infer relevant relationships from past history, the strategy-informed prompt appears to help them interpret and use those relationships more effectively when formulating intertemporal strategies. In addition, strategic delay and liquidity preservation for future obligations are substantially more common under the strategy-informed prompt.

Our third question concerns environments in which equilibrium theory does not provide a unique prediction. We retain the same basic two-agent indivisible-payment structure but change the cost and payment parameters so that the game admits multiple Nash equilibria. This treatment allows us to move beyond asking whether LLM agents can reproduce a unique theoretical benchmark and instead study how outcomes emerge through repeated interaction among adaptive agents. The sessions converge to different parts of the equilibrium set, and the decision records suggest that these differences reflect the co-evolution of strategies as each agent responds to the behavior generated by the other. Because the two agents rank the equilibria differently, equilibrium selection also determines how the costs of liquidity provision are distributed across participants.

The structured decision records help us investigate these outcomes in greater detail. Because the same types of economic considerations recur across rounds, we can track when agents report recognizing concepts such as liquidity recycling, future liquidity requirements, counterparty behavior, and the trade-off among liquidity, delay, and borrowing costs. We use these records primarily as a diagnostic measure rather than as evidence of the model's internal reasoning. They allow us to relate changes in stated strategic considerations to changes in observed actions, compare successful and unsuccessful convergence paths, and identify which relationships are absent when agents fail to reach equilibrium.

More broadly, our results suggest that evaluating autonomous AI systems requires studying an economy of agents rather than agents in isolation. As AI systems increasingly make operational decisions on behalf of organizations, their performance will depend not only on their individual reasoning capabilities but also on the strategic environment in which they operate, the information through which that environment is represented, and the behavior of other autonomous agents. Payment systems provide a transparent setting in which to study these issues, but the same questions arise wherever AI agents compete for resources, coordinate actions, respond to one another, or make interdependent decisions over time.

Related Literature. Our analysis relates first to the literature on strategic liquidity management in payment systems. Bech and Garratt (2003) characterize the strategic incentives associated with liquidity provision and payment delay, while subsequent analytical, simulation, experimental, and empirical work studies queuing, gridlock, liquidity-saving mechanisms, and participants' responses to payment-system design (Bech and Soramäki, 2001; Arciero et al., 2009; Galbiati and Soramäki, 2011; Abbink et al., 2017; Rivadeneyra and Zhang, 2022; Garratt et al., 2023; Alexandrova-Kabadjova et al., 2023). More recently, Castro et al. (2025) use reinforcement learning to estimate liquidity and payment-submission policies in stylized RTGS environments, while Aldasoro and Desai (2025) show that a single generative-AI agent can reproduce several canonical cash-management behaviors. Our focus is different: we study endogenous strategic interaction among multiple LLM agents and examine how equilibrium behavior depends on information design and equilibrium structure.

The paper also contributes to the emerging literature that uses generative AI as simulated economic agents and decision makers (Korinek, 2023, 2025; Hadfield and Koh, 2025; Anand et al., 2025; Gao et al., 2024; Kazinnik and Sinclair, 2025; Horton, 2023; Kazinnik, 2023) Much of this literature asks whether LLMs can reproduce individual behavior or perform economic decision tasks. Our experiments emphasize a different challenge: when autonomous agents interact, the relevant unit of evaluation is not only the individual decision maker but the system of strategically interacting agents. Individual competence does not by itself guarantee convergence, coordination, or efficiency. Outcomes can depend on what each agent understands about the environment, how agents form expectations about one another, and whether the game admits a unique strategic benchmark.

The rest of the paper proceeds as follows. Section 2 describes a stylized RTGS system. Section 3 describes the experimental design and LLM implementation. Section 4 discusses the experimental results. Section 5 interprets the broader implications of our results and discusses the limitations. Section 6 concludes and points out directions for future research.

2 The Payments System Environment

Our environment is a stylized RTGS payments system.[1] To settle transactions, participants in the RTGS system require liquidity provided by the central bank. Banks participating in the system use that liquidity to satisfy the exogenous and random payment demands they receive throughout the day from their (internal and external) clients.

Each participating bank can obtain the liquidity required to settle its payments either by pledging collateral to the central bank, typically at the start of the day, or by recycling liquidity received intraday from other banks. The former entails an opportunity cost, whereas incoming payments can

[1] In RTGS systems, payments are settled in real time and on a gross basis. As a result, offsetting payment obligations between two banks are not automatically netted, even when they are submitted at the same time and for the same value. In practice, however, many RTGS systems incorporate liquidity-saving mechanisms that allow queued payments to be offset bilaterally or multilaterally, reducing liquidity needs at the cost of some settlement delay (Martin and McAndrews, 2008; Alexandrova-Kabadjova et al., 2023).

Table 1: Timeline, decisions of the agent and constraints of the environment

Beginning of the day, $t = 0$	
Liquidity allocation:	ℓ_0
Cost of initial liquidity:	$r_c \cdot \ell_0$
Intraday periods, $t = 1, ..., T-1$	
Carryover payment demand balance:	U_{t-1}
New payment demand:	p_t
Decision to send:	$x_t \leq p_t + U_{t-1}$
Liquidity constraint:	$x_t \leq \ell_{t-1}$
Incoming payments:	R_t
Evolution of liquidity:	$\ell_t = \ell_{t-1} - x_t + R_t$
Evolution of unpaid balance:	$U_t = U_{t-1} + p_t - x_t$
Cost of delay:	$r_d \cdot U_t$
End-of-day borrowing, $t = T$	
Borrowing from central bank:	$B = \max\{0, U_{T-1} - \ell_{T-1}\}$
Cost of end-of-day borrowing:	$r_b \cdot B$

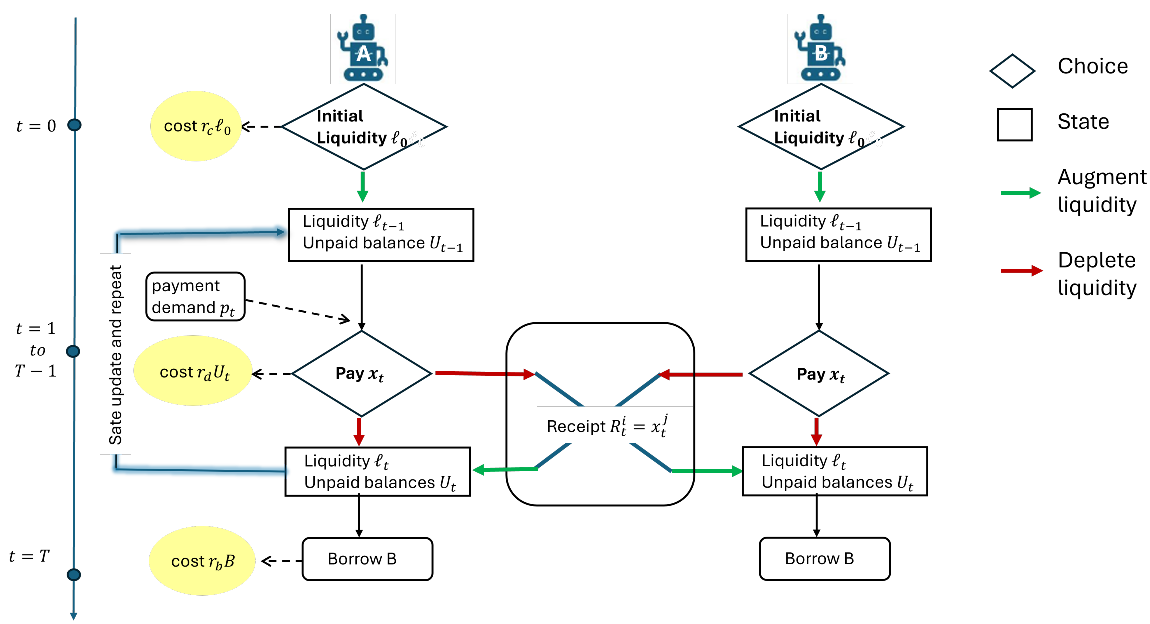


Figure 1: Simplified framework depicting two agents' initial iquidity and intraday payment decisions in stylized RTGS setup.

be reused without incurring additional liquidity costs. This creates an incentive to delay processing their own payments to wait for incoming payments. On the other hand, delaying payments is costly to banks if they don't service their clients in a timely fashion. Therefore, the bank's problem is to choose a policy that balances the cost of initial liquidity and the cost of delay by determining the level of initial liquidity and the timing of their own payments, all of which are conditional on the other bank's policies. At the end of the day, banks are required to satisfy all payment demands. If they do not have enough liquidity, they can borrow from the central bank at a cost.

Table 1 illustrates a typical day of a stylized RTGS system for each participant, and Figure 1 depicts the intraday liquidity and payments management for two cash management agents. In our simplified setup, each day is divided into intraday periods $t = 0, 1, 2, ..., T$. At the beginning of the day, $t = 0$, the agent allocates ℓ_0 as initial liquidity. Subsequently, at each intraday period, $t = 1, ..., T-1$, the agent receives payment demands p_t and chooses how much payment to send to other agents in that intraday period, denoted by x_t. The agent's choice is constrained by the available liquidity in that intraday period, so that $x_t \leq \ell_{t-1}$. The unpaid payment balance evolves according to $U_t = p_t + U_{t-1} - x_t$. At the end of each intraday period, the agent receives payments from other agents, which we denote by R_t. The available liquidity evolves according to: $\ell_t = \ell_{t-1} - x_t + R_t$.

The associated cost of the initial liquidity choice is $r_c \cdot \ell_0$ at a rate of r_c, and the cost of delay is given by $r_d \cdot U_t$ at a rate of r_d. If the agent does not have enough liquidity at $T-1$ to satisfy all the remaining payment demands, the agent can borrow from the central bank at a rate of r_b.

3 Experimental Design and Implementation

Using the general payment-system framework described above, we construct a stylized two-agent (A and B), four-period ($t = 0, 1, 2, 3$) cash-management game to study strategic interdependence. Both agents face the same payment profile (p_1, p_2) and the same cost parameters (r_c, r_d, r_b). Agent B faces an indivisible payment in the second intraday period ($t = 2$); other payments are divisible. Our game setup is similar to the Castro et al. (2025). It captures several core decisions in intraday liquidity management. Players make three interrelated choices: liquidity provisioning (how much costly liquidity to hold ex ante), payment timing (how much of the current obligation to settle immediately rather than delay), and liquidity buffering (whether to conserve available liquidity for future obligations). The game also embeds the key systems feature of strategic interdependence. One participant's outgoing payments become the other participant's inflows and thus their available liquidity. As a result, optimal decisions require forming beliefs about counterparties' liquidity positions and release strategies, and internalizing how current intraday release choices affect both (i) counterparties' ability to pay back and (ii) one's own liquidity and settlement outcomes in later intraday periods.

In the following, we describe the experimental design and implementation. We first introduce the three treatments and characterize the theoretical equilibrium benchmarks used to evaluate agent behavior. We then describe how the LLM agents are connected to the cash-management game and interact over repeated play. Next, we explain the experimental variation in information provided to the agents. Finally, we describe the structured decision rationales recorded alongside agents' choices.

3.1 Treatments

We construct three treatments based on the two-agent cash-management game with an indivisible payment. The treatments vary along two dimensions: the information provided to the LLM agents about the strategic relationships in the game and the equilibrium structure generated by the model parameters. Treatments 1 and 2 share the same game parameterization that leads to a unique equilibrium, but differ in the information supplied through the prompt. Treatment 1 provides strategy-informed instructions, whereas Treatment 2 provides only the rules and information needed to define the decision problem, leaving the agents to infer the relevant strategic relationships. Treatment 3 returns to the strategy-informed prompt as in Treatment 1 but uses a different parameterization that leads to multiple equilibria, allowing us to examine what emerges from the interaction of AI agents when equilibrium theory no longer selects a unique outcome. Table 2 summarizes the three treatments. We describe the information provision in detail in Section 3.4.

Table 2: Experimental treatments

Treatment	Payment profile (p_1, p_2)	Cost parameters (r_c, r_d, r_b)	No. of equilibria
1: Strategy-informed	(1.6, 2)	(0.8, 0.2, 0.5)	1
2: Rules-only	(1.6, 2)	(0.8, 0.2, 0.5)	1
3: Multiple equilibria	(2, 2)	(0.8, 0.2, 0.38)	4

3.2 Game paramterization and equilibria

We solve the subgame perfect Nash equilibrium (SPE), which serves as a useful benchmark for interpreting cash management game outcomes. Given initial liquidity choice, Agent A's best strategy is to pay as much as possible in both periods $t = 1, 2$. For Agent B, in period 2, the best strategy is as follows. If there is enough liquidity to pay the indivisible payment p_2, then p_2 is paid off, and the remaining liquidity is used to pay the carry-over divisible payment. Otherwise, the optimal strategy is to pay out the carry-over divisible payment as much as possible. Using this, we analyze the choice of (ℓ_0, x_1). In equilibrium, each agent's (ℓ_0, x_1) choice must be a best response to the other agent's strategy.

Regarding the parametrization, for Treatments 1 and 2, we set $p^A = p^B = [p_1, p_2] = [1.6, 2], r_c = 0.8, r_d = 0.2$, and $r_b = 0.5$. Under this parameterization, the unique SPE features strategic delay: in $t = 1$, Agent B retains a liquidity buffer to meet the indivisible payment p_2 in $t = 2$.[2] Table 3 reports the equilibrium initial liquidity (ℓ_0) and payment decision in $t = 1$ (x_1), along with each agent's total cost (C^A, C^B) and their combined cost ($C^A + C^B$). The unique theoretical equilibrium provides a clear benchmark for evaluating whether LLM agents can recognize the strategic structure of the environment and converge to the predicted outcome.

Table 3: Treatments 1 and 2: unique equilibrium

	ℓ_0^A	ℓ_0^B	x_1^A	x_1^B	C^A	C^B	$C^A + C^B$
Equilibrium	1.6	1.2	1.6	0.8	1.52	1.28	2.8

Notes: ℓ_0^i is initial liquidity for agent i; x_1^i is payment sent by agent i in $t = 1$; C^i is total cost for agent i.

For Treatment 3, we use the parameterization $p^A = p^B = [p_1, p_2] = [2, 2], r_c = 0.8, r_d = 0.2$, and $r_b = 0.38$. Under this parameterization, the game admits multiple theoretical equilibria, as reported in Table 4. This treatment allows us to investigate how the LLM agents navigate through such environments.[3] Note that the two agents rank the four equilibria differently. Agent A's ranking (according to lower cost) is 4, 3, 2, and 1, while Agent B has the reverse ranking. It is interesting to explore how agents navigate their conflicting preferences through interactions.

Table 4: Treatment 3: multiple equilibria

	ℓ_0^A	ℓ_0^B	x_1^A	x_1^B	C^A	C^B	$C^A + C^B$
Equilibrium 1	1.8	1.2	1.8	1	1.72	1.36	3.08
Equilibrium 2	1.6	1.4	1.6	1	1.64	1.52	3.16
Equilibrium 3	0.2	1.8	0.2	1.8	1.604	1.956	3.56
Equilibrium 4	0	2	0	2	1.56	2	3.56

Notes: ℓ_0^i is initial liquidity for agent i; x_1^i is payment sent by agent i in $t = 1$; C^i is total cost for agent i.

3.3 LLM agents implementation

To implement the agentic cash management game as described above, we connect two reasoning-capable LLM agents (ChatGPT 5.2 reasoning model) to payment games implemented in `oTree`, an open-source platform for interactive experiments, using a lightweight API interface. Figure 2

[2] Figure A.1 shows the ℓ_0 component of the best response functions.

[3] A classic motivation for laboratory experiments with human participants is to study equilibrium selection and the behavioral mechanisms through which one equilibrium is selected over another in environments with multiple theoretical equilibria.

summarizes the high-level implementation framework, along with the game information provided to the agents, the action choices, and the output data generated after each action, which includes both `oTree` game outputs and LLM reasoning traces.

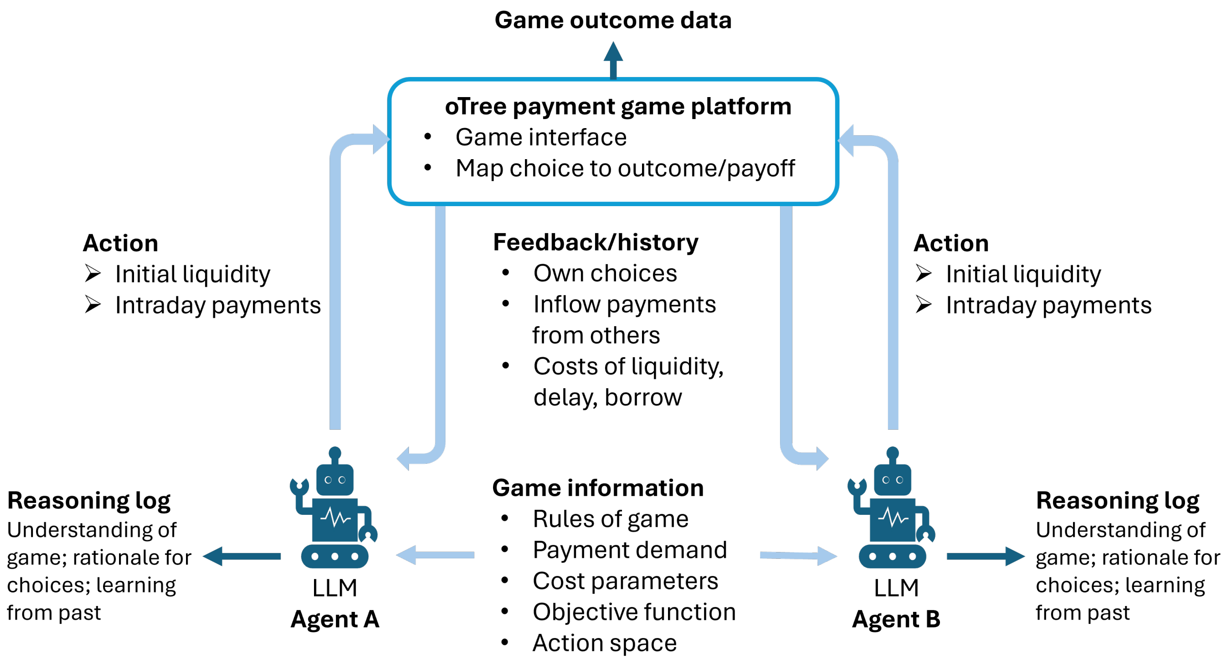


Figure 2: Implementation Framework for Agentic Cash Managers

We choose a reasoning-capable LLM because the task requires algebraic manipulation and multi-step logical reasoning, including tracking state variables across periods, applying payment rules, and evaluating cost trade-offs.[4]

For each treatment, we conduct five independent sessions in which two LLM agents interact repeatedly. Each session consists of 50 repetitions, or *rounds*, of the cash-management game. Within each round, the agents first choose initial liquidity and subsequently make intraday payment decisions according to the sequence described in Section 3.2. Each round starts from the same exogenous payment profile, while agents receive information about outcomes from previous rounds. Repetition allows the LLM agents to adapt their decisions to observed counterpart behavior and past outcomes, paralleling the repeated-game design commonly used in experiments with human subjects.

Before the first round of a session, each LLM agent receives a set of experimental instructions describing its role and objective, the structure and rules of the cash-management game, and the information available when making decisions. These instructions provide the common context within which the agents interact throughout the session. Depending on the treatment, the experimental

[4] At this stage, we use the model with its default parameters. The model is configurable along several dimensions, including reasoning capability.

instructions contain or omit strategic information across treatments, which we describe in detail in Section 3.4. See Appendix B for the full text of the experimental instructions.

We implement the repeated interaction in `oTree`, an open-source platform for dynamic and interactive experiments. Each round is implemented as a sequence of decision pages corresponding to the stages of the cash-management game. The program maps agents' choices into subsequent states, outcomes, and costs, carries the resulting variables forward across stages, and synchronizes the two agents when their decisions are simultaneous. All parameters, decisions, and outcomes are recorded in a structured database, which is used both to construct the history provided to the agents in subsequent rounds and for analysis after the session.

3.4 Information and Prompt Conditions

Prompt and context design can be an important component of introducing AI agents, particularly when general-purpose LLMs are adapted to specialized decision-making tasks as we do in this paper. Unlike models trained specifically for a particular environment, their decisions are conditioned on rich contextual information provided through the prompt.[5] Prompt design can also affect which relationships in the environment become salient to the agent and are incorporated into its decisions. This is particularly important in strategically interactive settings, where successful behavior may require recognizing indirect relationships between one agent's actions and the future actions of others. If alternative representations of the same decision environment generate different behavior, prompt design may therefore be not only a performance consideration but also an important element in the governance of autonomous AI systems.

We provide three types of context: role and objectives, games and state information, and strategic information. The first two types of information are provided in all three treatments. The third type of information is present in Treatments 1 and 3 and omitted in Treatment 2.

Type 1 information defines the high-level LLM's role as a cash-management agent operating in a repeated RTGS payment system game (see Box 1). It specifies the agent's objective, decision environment, and requirement to minimize total cost through liquidity and payment choices while using information from previous rounds. It also imposes a structured-output constraint by requiring all responses to be valid JSON with predefined keys.

The second type of context is game and state information, which includes the rules, variables, costs, current state, and history. Given the LLM models' algebraic and natural language capabilities, we combine text, simple equations, and compact numerical summaries to describe the environment. We start by defining the variables and the cost parameters, then we describe the game flow with a series of bullet points. At each decision point, the agent is provided with (i) the rules of the game

[5] This highlights a key difference from the classical RL approach used in Castro et al. (2025). In that setting, agents, represented as neural network learn from numerical state variables and an explicit reward signal.

(e.g., RTGS constraints, liquidity constraints, other players participation, etc.), (ii) the current state (e.g., liquidity balances, outstanding/queued payment requests, remaining periods, information about expected inflows, and history), and (iii) an explicit objective: minimize total cost, defined as a weighted combination of liquidity costs and delay costs (and, when applicable, borrowing/settlement costs), subject to feasibility constraints. In addition, to carry information across rounds, the decision page displays the history of selected variables from all previous rounds: initial liquidity, liquidity balance at the end of the period, payment in each period, payment receipts or inflows in each period, outstanding balances in each period, cost of liquidity, cost of delays, cost of borrowing, and total cost.

The third type of context is strategic information, including qualitative explanations of strategically relevant relationships and guidance. For example, a central feature of the payment system is liquidity interdependence: as one agent's outflow becomes another agent's inflow, each agent's available liquidity evolves with strategic inflows generated by the other agent's payment decisions. In addition, an agent's current outflow may affect its own future inflow as liquidity is recycled across agents and periods.

Box 1: Role and objective prompt sample (present in all treatments)

"You are a cash management agent in a large-value payment system to manage liquidity and payments in a series of four-period games. You are playing the game with another cash management agent. Each four-period game is called a round. Your goal is to minimize the total cost in each round by optimizing liquidity and payment decisions while learning from previous rounds. Always respond in valid JSON with two keys:

- **Reasoning**: a concise justification of your choice, focusing on strategic choices. Explain the steps when making algebraic calculations.
- **Answer**: your final decision (e.g., "Yes" to confirm understanding)."

Box 2: Game and state information prompt sample (present in all treatments)

In each experiment, acting as a cash manager, you will play a repeated multi-agent cash-management game with a specified payment profile. At the beginning of each round, you choose an initial liquidity level, which is costly but can be used to settle payment obligations in subsequent periods. In each intraday period, you observe your outstanding obligation and available liquidity, and then choose how much to pay. At the end of each intraday period, you receive an inflow determined by the other agent's payment decision. Unpaid amounts are carried forward and incur delay costs, while any remaining obligation at final settlement requires costly borrowing. Your objective is to minimize total round cost, defined as the sum of initial liquidity cost, delay cost, and final borrowing cost.

Box 3: Strategic information prompt sample (present in Treatments 1 and 3)

"**Interaction with the Other Agent**

- Your inflows R_1 and R_2 depend on the other agent's liquidity and payment decisions.
- Conversely, your payments affect the other agent's inflows and available liquidity.
- This creates reciprocity across periods. In particular, your period-1 payment, x_1, can affect your own inflow in period 2, R_2, through the following mechanism:
  1. Period 1: the payment you send, x_1, becomes the other agent's inflow R_1, increasing their liquidity carried into period 2 (denoted ℓ_1).
  2. Period 2: if that additional liquidity relaxes the other agent's liquidity constraint, they may be able to pay you more, so part or all of x_1 can come back to you as higher R_2. If the other agent is not liquidity-constrained, sending extra in period 1 will not change your R_2."

For the agent who faces a lumpy payment, we remind it "Do not ignore strategies to keep part of ℓ_0 in period 1 to ensure the payment of indivisible p_2 in period 2."

3.5 Reasoning texts

Besides the information provided, we also prompt the LLM agents to record the reasoning texts. This serves several purposes. First, they provide a diagnostic tool for identifying whether the agents correctly interpret the rules, constraints, and strategic interdependencies of the game. This information can be used to refine the experimental instructions and reduce ambiguity in the representation of the decision environment. Second, the recorded reasoning logs help researchers follow the evolution of strategic behavior across rounds, including how agents use historical outcomes, revise expectations about counterparties, and adjust their liquidity and payment decisions. Third, although these texts should not be interpreted as direct representations of the models' internal computational processes, they provide an auditable record of the considerations reported by the agents at the time of each decision. Such records may be useful in regulated or operational settings where documentation, review, and traceability of AI-supported decisions are important.

4 Results

Figures 3 to 5 show the results of the three treatments. Each figure shows the time paths of the two agents' initial liquidity ℓ_0, payment in $t = 1$, x_1, and total cost C for each of the five independent sessions. We organize the discussion of the results around a series of findings. We first investigate Treatment 1 to evaluate the LLM agents' ability to engage in nontrivial strategic deliberations when the prompts involve explanations of strategically relevant relationships. We then compare Treatments 1 and 2 to identify the effects of information and prompt conditions. Finally, we examine how the LLM agents function in an environment that involves multiple theoretical equilibria.

4.1 Strategic capabilities of LLM agents

We first investigate whether general-purpose LLM agents are capable of nontrivial strategic deliberation in the specialized cash-management environment through rich context conditioning, as implemented in Treatment 1. From Figure 3, all five sessions of Treatment 1 converge to the unique SPE within 20 rounds.

In earlier rounds, agents try different strategies and learn about the other agent's behaviors and patterns of incoming funds (denoted by R). Agent A has only divisible payments, and it is a dominant strategy to pay as much as possible. This strategy is always followed. Agent A's initial liquidity choice is quite stable: it sets $\ell_0 = x_1 = 1.6$ almost always and experiments with a lower ℓ_0 only in a few rounds. Agent B has a lumpy payment in period 2 and faces more strategic payment decisions, experimenting more in the initial rounds of a session.

We can construct Agent B's strategy from the reasoning log. To illustrate how the agents adapt across rounds, consider Agent B in Session 2. Its behavior can be summarized in three stages. *First, in the initial exploration stage (Rounds 1–3),* the agent experiments with alternative liquidity levels while learning about incoming payments. With no history in Round 1, it initially assumes no inflows and chooses sufficient liquidity to settle its first-period payment. After observing positive inflows, it subsequently experiments with both higher and lower liquidity. In Round 3, it sets $\ell_0 = x_1 = 0$ based on the expectation that the previously observed inflows will continue, but this strategy results in higher delay and borrowing costs.

Second, in the strategic-learning stage (Rounds 4–9), Agent B begins to recognize the relationship between its period-1 payment x_1 and its subsequent inflow R_2, as well as the importance of the counterparty's period-1 payment R_1 for financing the indivisible payment in period 2. It therefore experiments with combinations of ℓ_0 and x_1 that preserve a liquidity buffer while adapting to variation in the observed R_1. These experiments generate temporary deviations from the equilibrium strategy as the agent updates its expectations about the counterparty's behavior.

Finally, in the convergence stage (Round 10 onward), Agent B uses the accumulated history to recognize that $R_1 = 1.6$ is the typical inflow and chooses its best response to this pattern. It selects $(\ell_0, x_1) = (1.2, 0.8)$, which leaves sufficient liquidity to settle the indivisible period-2 payment while avoiding unnecessary initial-liquidity and borrowing costs. From Round 11 onward, the agent maintains this strategy, which coincides with its equilibrium strategy.

Reasoning of LLM: treatment 1

Session 2, Agent B, round 10:

From rounds 1–9, R_1 is usually 1.6 (7/9 times; average ≈ 1.51), so I target the low-cost plan that works when $R_1 \approx 1.6$ without leaving idle liquidity.

If I set $\ell_0 = 1.2$, then in period 1 I can choose $x_1 = 0.8$ (as in round 4): $U_1 = p_1 - x_1 = 1.6 - 0.8 = 0.8$ and $\ell_1 = \ell_0 - x_1 + R_1 = 1.2 - 0.8 + 1.6 = 2.0$, so the indivisible $p_2 = 2$ can be paid in period 2. With p_2 paid, unpaid after period 2 is $U_2 = 3.6 - \ell_0 - R_1 = 3.6 - 1.2 - 1.6 = 0.8$.

Using the observed pattern $R_2 \approx x_1 = 0.8$, borrowing is $B = \max(0, 3.6 - \ell_0 - R_1 - R_2) = \max(0, 3.6 - 1.2 - 1.6 - 0.8) = 0$.

Total cost: $TC = 0.8 * \ell_0 + 0.2 * (U_1 + U_2) + 0.5 * B = 0.8 * 1.2 + 0.2 * (0.8 + 0.8) + 0 = 0.96 + 0.32 = 1.28$, which is lower than the typical $\ell_0 = 1.0$ outcome (≈ 1.32) and avoids overpaying liquidity cost seen with higher ℓ_0.

The top two panels of Figure A.2 show the word clouds for the reasoning text, separately for the two agents. The box "Reasoning of LLM: Treatment 1" shows the reasoning log for Agent B in round 10 of session 2. From the reasoning log, we see that the LLM agents understand the key concepts and tradeoffs of the cash management game. They use algebra and consult history to derive their decisions. Words such as "cost", "delay," "liquidity," and "borrowing" are frequently mentioned. Words like "consistently," "stable," and "pattern" suggest that agents infer patterns from past outcomes. Agents also understand the interdependence of fund flows and often refer to incoming payments R_1, R_2, and words such as "inflow" and "reciprocity" appear frequently too. Agents even understand the more intricate concept of liquidity recycling and that their own payments can influence their incoming flow in future periods, and their initial liquidity often uses the relationship $x_1 = R_2$. In addition, Agent B fully understands the indivisibility of p_2 payment: the word "indivisible" stands out.

We summarize the above discussion as Finding 1:

Finding 1. *While prompted with rules of game and explanations about strategic relationships, general purpose LLM agents combine strategic information with feedback from repeated interaction to implement the nontrivial equilibrium liquidity-management strategy.*

The agents do not receive the equilibrium actions or an analytical solution directly. Instead, they combine the strategic information provided in the instructions with feedback from previous rounds to refine their liquidity and payment decisions. The baseline treatment does not identify the separate contribution of these elements to convergence. We examine one component—the provision of explicit strategic information—in the next subsection.

4.2 Role of strategy prompting

Treatments 1 and 2 allow us to study how the representation of a specialized decision environment affects the behaviors of general-purpose LLM agents. Across the two treatments, agents receive the same role, objective, game rules, state variables, cost parameters, and history of previous play. The treatments differ only in whether the prompt additionally provides explicit descriptions of strategically relevant relationships and strategy guidance. This distinction allows us to separate an agent's ability to act strategically when relevant mechanisms are explicitly described from its ability to infer those mechanisms from the formal rules and its own experience.

From Figures 3 and 4, all five strategy-informed sessions converge to the SPE, compared with only one of the five rules-only sessions. Presenting the strategy information, therefore, has a substantial impact on the performance of the LLM agents, as summarized in the finding below.

Finding 2. *Equilibrium convergence is highly sensitive to how the strategic environment is represented in the prompt. Strategy-informed prompting is associated with more frequent recognition and use of strategically relevant relationships and substantially more equilibrium-consistent behavior.*

We can gain insights into the reason by examining the reasoning text. The word clouds in Figure A.2 suggest a piece of quick evidence: "reciprocity" (matched on `reciproc*`) is prominent in Treatment 1 but not in treatment 2. Table 5 summarizes differences in Agent B's reported strategic considerations and behavior across the strategy-informed and rules-only treatments. The first two rows capture recognition of the dynamic consequences of liquidity recycling, while the last two rows capture whether the agent preserves liquidity for the future indivisible payment and implements the associated strategic delay.

The reasoning records suggest two distinct effects of the strategy-informed prompt. First, agents in both treatments can infer from the common history that the period-1 payment x_1 is related to the subsequent inflow R_2: the relation $R_2 \approx x_1$ is mentioned in 44% of rounds in Treatment 1 and 33% in Treatment 2. The larger difference in mentions of "reciprocity" (21% versus 3%) should be interpreted cautiously because that term is introduced explicitly in the Treatment 1 prompt. Rather than treating the word frequency itself as evidence of deeper reasoning, we interpret the combination of recognizing $R_2 \approx x_1$ and subsequent behavior as evidence that the strategy-informed prompt helps agents understand and use the dynamic consequences of liquidity recycling.

Second, the strategy-informed prompt explicitly reminds Agent B not to ignore strategies that preserve liquidity for the future indivisible payment. Consistent with this guidance, Agent B engages in strategic delay, $\ell_0 > x_1$, in 90% of rounds in Treatment 1 compared with 51% in Treatment 2, and more often reports keeping liquidity for p_2 (14% versus 3%). Together, these patterns suggest that strategy prompting improves performance by helping the agent connect observed relationships in the history to the intertemporal strategy required for equilibrium play.

Table 5: Strategic Reasoning and Liquidity-Buffering Behavior in Treatments 1 and 2

B's action or mention	treat 1	treat 2
mention reciprocity	21%	3%
mention $R_2 \approx x_1$	44%	33%
engage in strategic delay ($\ell_0 > x_1$)	90%	51%
mention keep liquidity for p_2	14%	3%

4.3 Environment with multiple equilibria

Treatment 3 examines a distinctly multi-agent aspect of the problem: what happens when strategically capable agents interact in an environment in which theory does not select a unique outcome. In Treatments 1 and 2, the unique equilibrium provides a single benchmark toward which successful learning can converge. With multiple equilibria, by contrast, an agent's appropriate strategy depends on the strategy that evolves on the other side, so the long-run outcome must emerge jointly through repeated interaction.

The four equilibria in Treatment 3 also differ substantially in how costs are distributed across the two agents. Agent A prefers the equilibria in the order $4,3,2,1$, from lowest to highest cost, whereas Agent B has exactly the reverse ranking, $1,2,3,4$. Thus, equilibrium selection is not simply a matter of coordinating on a commonly preferred outcome. Different patterns of interaction can lead the agents toward equilibria that favor one agent or the other to different degrees. This feature makes the treatment useful for examining how the strategies of autonomous agents co-evolve when each agent adapts to the behavior generated by the other.

The time paths of the five sessions are shown in Figure 5. Overall, the sessions tend to converge to a particular outcome, as summarized in Table 4. The only exception is session 2, which goes to the neighborhood of equilibria 3 and 4; however, the two equilibria are very close to each other.

Finding 3. *When equilibrium is non-unique, independent LLM-agent sessions tend to select different theoretical equilibria as long-run outcomes.*

As shown in Table 6, the sessions result in different allocations. Combining with Table 4, session 3 converges to equilibrium 1, an outcome where A provides more initial liquidity and B has a lower cost. By contrast, the other sessions end up with outcomes where B provides the majority of initial liquidity and A enjoys a lower cost. Because the agents, prompts, game parameters, and initial information are held fixed across sessions, the variation in long-run outcomes arises through the realized sequence of interaction and adaptation. As each agent observes the other's choices, updates

Table 6: Long-run outcomes in the multiple-equilibrium treatment

Session	Long-run outcome	
1	$(\ell_0^A, x_1^A) = (0.2, 0.2)$, $(\ell_0^B, x_1^B) = (1.8, 1.8)$	Equilibrium 3
2	In the neighborhood of Equilibria 3 and 4	Near Equilibrium 3/4
3	$(\ell_0^A, x_1^A) = (1.8, 1.8)$, $(\ell_0^B, x_1^B) = (1.2, 1.0)$	Equilibrium 1
4, 5	$(\ell_0^A, x_1^A) = (0, 0)$, $(\ell_0^B, x_1^B) = (2, 2)$	Equilibrium 4

Notes: The table reports the long-run behavior of the five independent sessions in Treatment 3. Equilibrium numbering follows Table 4. Session 2 settles in the neighborhood of Equilibria 3 and 4 rather than converging precisely to either equilibrium.

its expectations, and adjusts its own strategy, these adjustments change the environment faced by the counterparty in subsequent rounds. The resulting co-evolution of strategies can therefore place otherwise identical sessions on different paths toward equilibrium.

The decision records suggest that equilibrium selection emerges through feedback between the agents' evolving policies. Each agent forms expectations about future inflows from realized history, but these inflows are themselves generated by the counterparty's current strategy. An adjustment by one agent, therefore, changes the decision environment subsequently faced by the other. In Session 3, Agent A learns that reducing its period-1 payment can sharply reduce its subsequent inflow and consequently sustain a high-payment strategy; Agent B, receiving substantial early liquidity from A, learns to retain only the buffer needed for its future indivisible payment. The pair converges to Equilibrium 1. In Sessions 1, 2, 4, and 5, by contrast, higher early payments by Agent B provide liquidity to Agent A and reduce A's incentive to acquire costly initial liquidity. As A reduces its early payment, B receives less incoming liquidity and responds by relying more heavily on its own liquidity, reinforcing the movement toward the equilibria 3–4 region that favors A.

These differences are economically consequential because the two agents rank the equilibria in opposite orders. Thus, the interaction determines not only whether strategically coherent behavior emerges, but also which agent bears the cost of providing liquidity.

This result highlights a feature of multi-agent AI systems that is absent from stand-alone evaluations. Even when individual agents are capable of strategically coherent behavior and ultimately play mutual best responses, the interaction among them need not generate a unique outcome. Which equilibrium emerges—and therefore how the gains and costs of interaction are distributed across agents—can itself be an outcome of the learning and adaptation process.

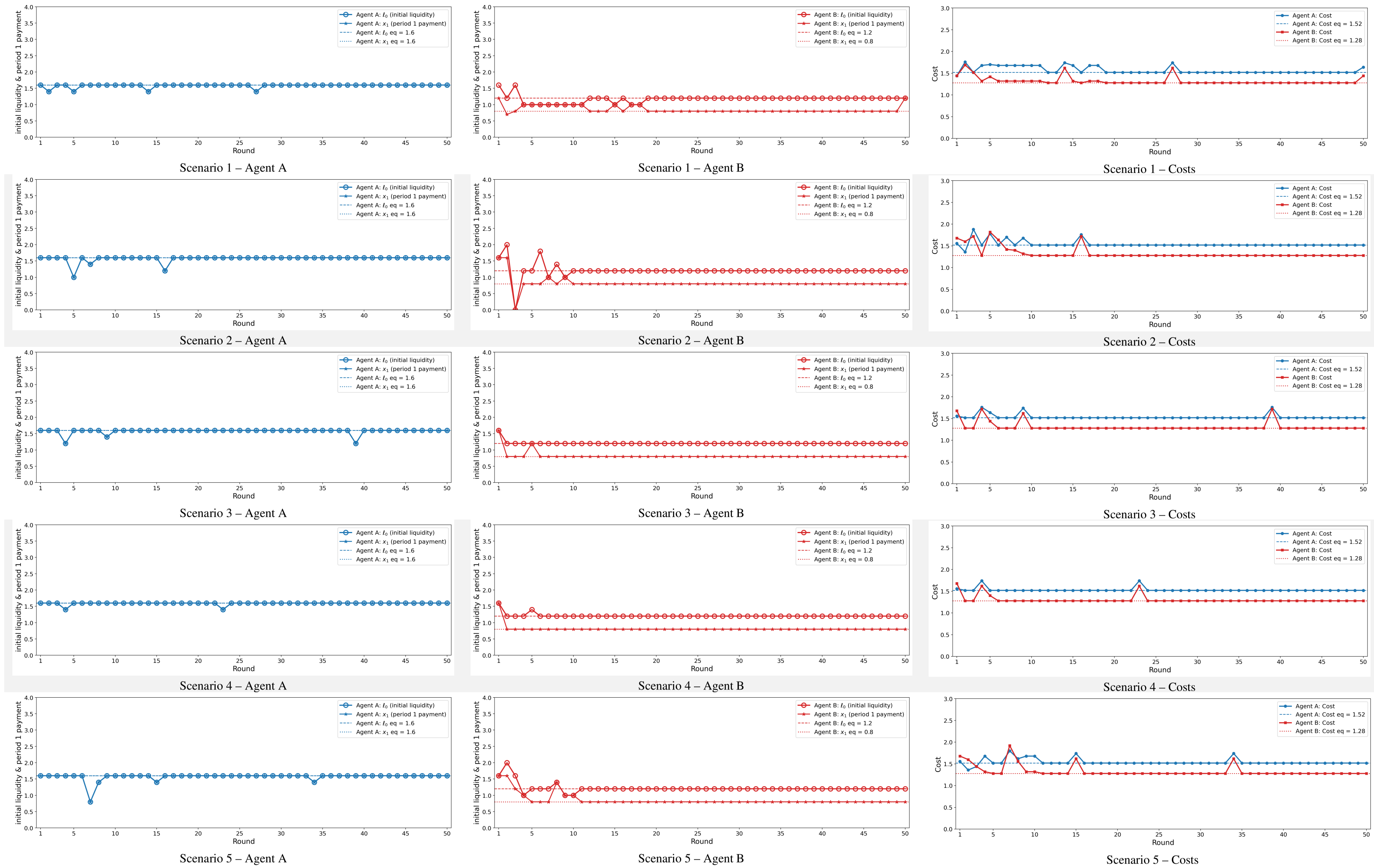


Figure 3: Decision paths and cost outcomes for Treatment 1.
Each row reports Agent A decisions, Agent B decisions, and the corresponding cost profile.

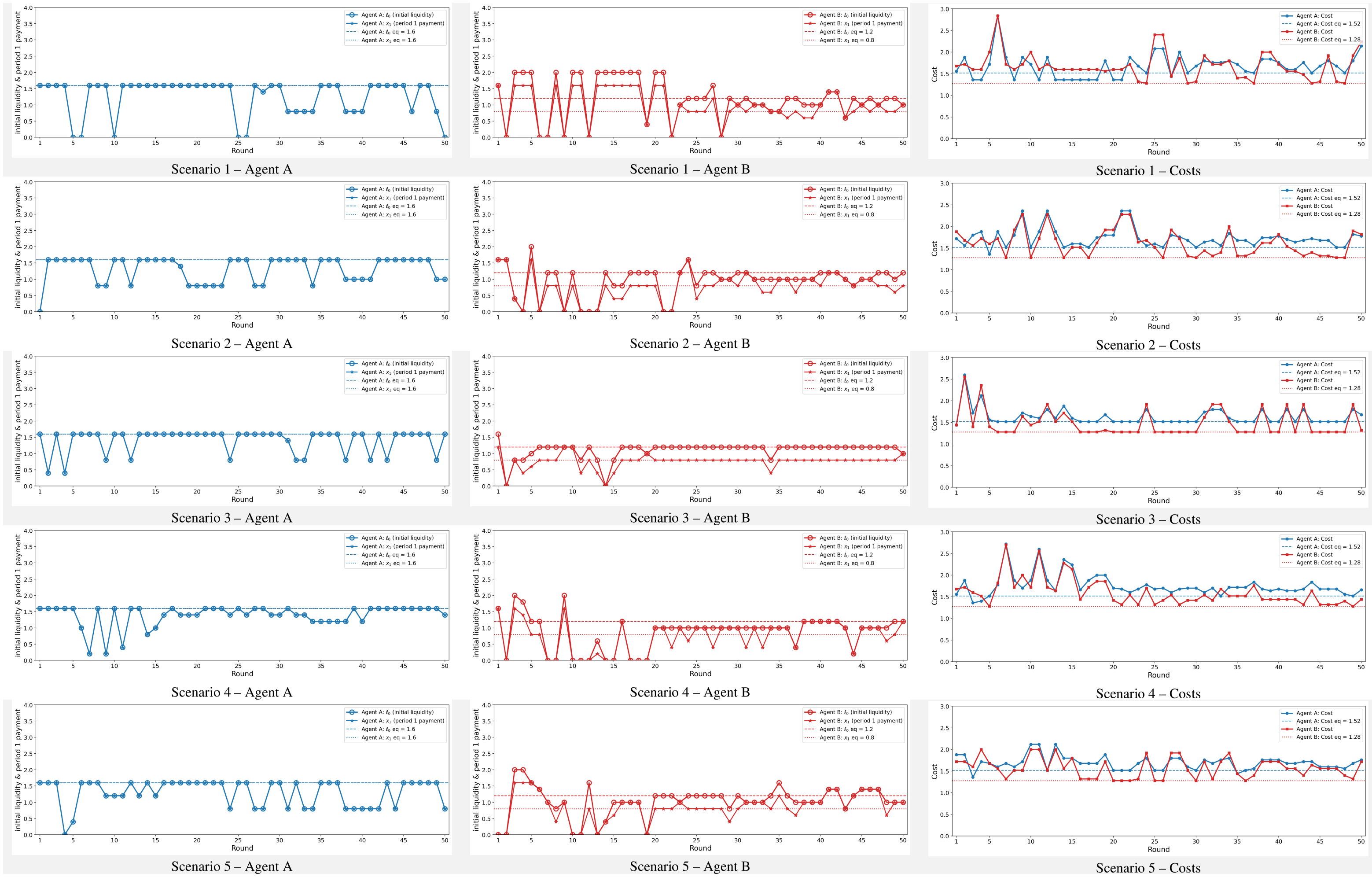


Figure 4: Decision paths and cost outcomes for Treatment 2.
Each row reports Agent A decisions, Agent B decisions, and the corresponding cost profile for a given scenario.

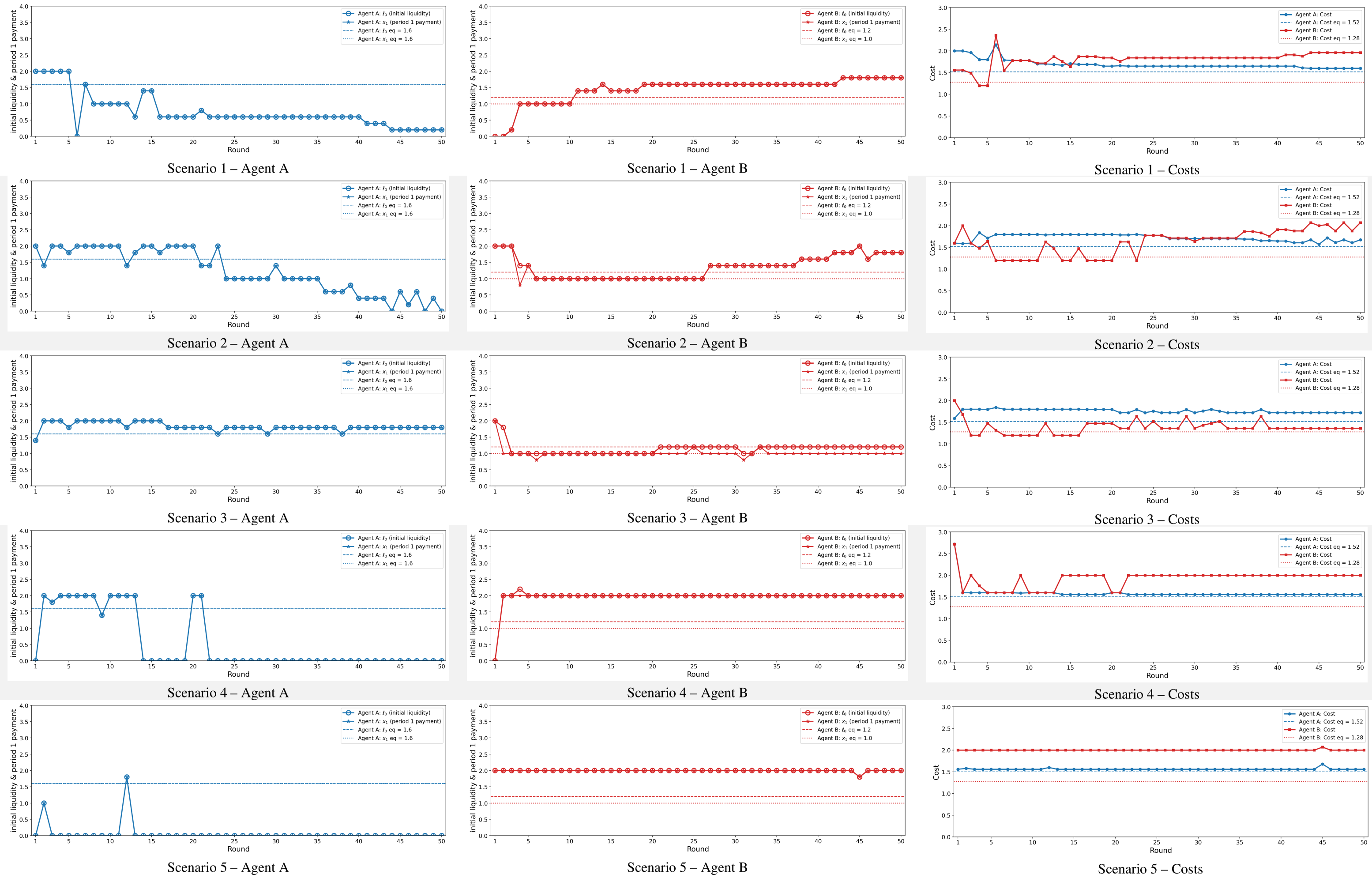


Figure 5: Decision paths and cost outcomes across five lumpy-payment scenarios with parameterization in Table 4. Each row reports Agent A decisions, Agent B decisions, and the corresponding cost profile for a given scenario.

4.4 Supplementary experiments

We conduct two supplementary experiments to examine whether the behavior observed in the focal two-agent indivisible-payment environment extends to other cash-management settings. Here we summarize the main results, and details are reported in Appendix C.

The first considers a simpler two-agent game with divisible payments. In this environment, all five sessions converge rapidly to the theoretical equilibrium, with agents learning to rely on liquidity recycling to economize on initial liquidity. This treatment also provides a benchmark for comparison with the reinforcement-learning agents studied by Castro et al. (2025).

The second supplementary experiment extends the framework to three interacting agents. One agent has insufficient liquidity to pay both counterparties immediately and must decide which payment to prioritize. The agents learn to direct scarce liquidity toward the counterparty that subsequently returns liquidity, showing that the framework extends beyond bilateral interaction to counterparty-specific payment decisions.

5 Discussion

Our experiments use intraday liquidity management in an RTGS system as a controlled environment to study strategic interaction among AI agents. Our results have broader implications for the use of AI agents in interactive financial systems. Many financial environments—including electronic trading, repo and secured funding markets, securities settlement and collateral management, and large-value payment systems—involve strategic interdependence, such that one institution's decisions affect the states, opportunities, or incentives faced by others. The FSB has progressively expanded its work on AI in finance from financial-stability implications (Financial Stability Board, 2024), to monitoring adoption and related vulnerabilities (Financial Stability Board, 2025), and more recently to sound practices for responsible AI adoption (Financial Stability Board, 2026). We discuss three aspects: the importance of system-level evaluation, the role of prompt and context design, and the potential value of structured decision records for monitoring and diagnosis.

5.1 Evaluation of interacting AI agents

Evaluating an AI agent on individual decision tasks may be insufficient when its decisions are strategically interdependent with those of other agents. In such settings, evaluation should also consider the outcomes that emerge through interaction. Our multiple-equilibrium treatment illustrates this distinction. Independent sessions converge to different parts of the theoretical equilibrium set, even though the agents, prompts, and economic environment are held fixed. The decision records suggest that these differences arise through the co-evolution of strategies: each agent adapts to the behavior of its counterparty, thereby changing the environment to which the counterparty subsequently responds. Strategically coherent behavior at the individual level therefore need not imply a unique system-level outcome.

This observation motivates a role for multi-agent simulation in evaluating autonomous financial systems. Such evaluation can examine not only whether individual agents respond appropriately to incentives and constraints but also how interacting agents jointly adapt and which outcomes emerge under alternative counterpart behaviors, institutional parameters, information structures, and initial conditions. This is particularly important when multiple equilibria differ in efficiency or distributional consequences. In our setting, the two agents rank the equilibria in opposite orders, so equilibrium selection determines not only the pattern of liquidity provision but also how its costs are distributed across participants.

5.2 Prompt and context design

Our results also highlight the importance of how a specialized financial decision problem is represented to a general-purpose LLM. The strategy-informed and rules-only treatments provide the same objective, economic environment, action space, and realized history, but differ in whether strategically relevant relationships are explicitly described. The substantial difference in equilibrium convergence shows that formally sufficient information need not be sufficient for an agent to identify and use the relevant strategic relationships effectively.

Prompt and context design should be an integrated part of an autonomous decision system rather than simply a performance-tuning exercise. Instructions should be documented and tested. After the introduction of AI agents, the instructions should be version controlled and subject to formal approval when modified.

5.3 Decision records, monitoring, and auditability

The structured decision rationales provide information that is not contained in actions and outcomes alone. They allow us to examine which strategic relationships agents report considering, how their expectations evolve, and whether changes in these considerations accompany changes in behavior. In our experiments, these records help distinguish, for example, whether agents recognize liquidity recycling and the need to preserve liquidity for future obligations.

Although these records cannot be interpreted as direct observations of the model's internal computational process, they provide a contemporaneous and auditable account of the considerations reported by the agent. In operational applications, such records could complement outcome-based monitoring by helping to diagnose unexpected behavior, compare model or prompt versions, and identify changes in how an agent represents key features of its environment. They should complement, rather than replace, direct monitoring of actions, constraints, and outcomes.

5.4 Limitations

Our experiments are deliberately stylized, and several features limit the scope of the conclusions. We study a single reasoning-capable LLM and a limited number of independent sessions, so the results need not generalize to models with different capabilities or configurations. The agents also operate in a small, well-defined environment with accurate state information, complete histories, and common objectives; real financial systems involve greater heterogeneity, imperfect information, and substantially more complex interactions. Finally, our prompt experiment examines one particular contrast between strategy-informed and rules-only instructions and therefore does not establish robustness to other forms of prompt or context design.

6 Conclusion and Future Research

We study strategic interaction among general-purpose LLM agents in a stylized RTGS liquidity-management environment. The experiments show that when provided with sufficiently rich contextual information, LLM agents can implement nontrivial strategic policies, adapt to counterpart behavior through repeated interaction, and converge to theoretically predicted outcomes. At the same time, their performance depends on how the decision environment is represented in the prompt. The structured decision rationales generated alongside their actions provide an additional record that can be used to diagnose how agents interpret strategic relationships and why their behavior changes over time.

Beyond demonstrating these capabilities in a stylized setting, our experimental framework provides a foundation for studying AI agents in richer and more realistic financial-system environments. For RTGS systems, the framework developed can serve as a prototype to guide the construction of more realistic systems. It can be extended to study richer features such as heterogeneous participants, payment prioritization, delay propagation, gridlock, liquidity-saving mechanisms, stress conditions, and larger payment networks. More realistic experiments could incorporate empirical payment-flow patterns and institution-specific characteristics (for example, we can use RTGS data to create bank personas to act as cash managers in an RTGS simulator). The AI-integrated simulator can be used to understand the benefits, challenges, limitations, and risks of AI integration. It also provides a safe, transparent sandbox to test policies, system designs, and the systemic implications of AI integration. RTGS participants can also use an AI-integrated simulator to train new cash managers, simulate stress testing, and improve operational efficiency as market conditions evolve and liquidity becomes scarcer in the payment system.

The same approach can also be applied to other financial environments in which decisions are strategically interdependent, including electronic trading, secured funding markets, and securities settlement and collateral management. Controlled multi-agent simulations can therefore complement conventional model- and task-level evaluation by examining how AI agents respond to one another, how outcomes depend on instructions and institutional design, and how behavior changes under alternative market and stress scenarios. Such evidence can help inform emerging governance and regulatory approaches to AI adoption in finance, including requirements for pre-deployment testing, documentation of agent instructions, monitoring of interacting agents, and traceability of automated decisions. In this sense, the framework developed here provides both a research laboratory for studying strategic AI behavior and a potential sandbox for evaluating how autonomous financial agents can be introduced safely and effectively.

References

Abbink, K., R. Bosman, R. Heijmans, et al. (2017). Disruptions in large value payment systems: An experimental approach. *International Journal of Central Banking 13*(4), 63–95.

Aldasoro, I. and A. Desai (2025). AI agents for cash management in payment systems. Staff Working Papers 35, Bank of Canada.

Alexandrova-Kabadjova, B., A. Badev, S. Benchimol Bastos, E. Benos, F. Cepeda-Lopéz, J. Chapman, M. Diehl, I. Duca-Radu, R. Garratt, R. Heijmans, A. Kosse, A. Martin, T. Nellen, T. Nilsson, J. Paulick, A. Pustelnikov, F. Rivadeneyra, M. Rubem do Coutto Bastos, and S. Testi (2023). Intraday liquidity around the world. Technical Report 1089, BIS Working Paper.

Anand, K., S. Kazinnik, A. Leonello, and E. Panetti (2025). Ex machina: Financial stability in the age of artificial intelligence. *Available at SSRN 5543139*.

Arciero, L., C. Biancotti, L. D'Aurizio, and C. Impenna (2009). Exploring agent-based methods for the analysis of payment systems: A crisis model for StarLogo TNG. *Journal of Artificial Societies & Social Simulation 12*(1).

Bech, M. L. and R. Garratt (2003, April). The intraday liquidity management game. *Journal of Economic Theory 109*(2), 198–219.

Bech, M. L. and K. Soramäki (2001). Gridlock resolution in interbank payment systems. Technical Report 9/2001, Bank of Finland.

Castro, P., A. Desai, H. Du, R. Garratt, and F. Rivadeneyra (2025). Estimating policy functions in payment systems using reinforcement learning. *ACM Transactions on Economics and Computation 13*(1), 1–31.

Financial Stability Board (2024, November). The financial stability implications of artificial intelligence. Technical report, Financial Stability Board.

Financial Stability Board (2025, October). Monitoring adoption of artificial intelligence and related vulnerabilities in the financial sector. Technical report, Financial Stability Board.

Financial Stability Board (2026, June). Sound practices for responsible adoption of artificial intelligence (AI): Consultation report. Technical report, Financial Stability Board.

Galbiati, M. and K. Soramäki (2011). An agent-based model of payment systems. *Journal of Economic Dynamics and Control 35*(6), 859–875.

Gao, C., X. Lan, N. Li, Y. Yuan, J. Ding, Z. Zhou, F. Xu, and Y. Li (2024). Large language models empowered agent-based modeling and simulation: a survey and perspectives. *Humanities and Social Sciences Communications 11*, 1259.

Garratt, R. J., Z. Lu, and P. Tian (2023). How banks create gridlock to save liquidity in canada's large value payment system. Technical Report 2023-26, Bank of Canada.

Hadfield, G. K. and A. Koh (2025). An economy of AI agents. arXiv preprint.

Horton, J. J. (2023). Large language models as simulated economic agents: What can we learn from homo silicus? Technical report, National Bureau of Economic Research.

Kazinnik, S. (2023). Bank run, interrupted: Modeling deposit withdrawals with generative AI. *Available at SSRN*.

Kazinnik, S. and T. M. Sinclair (2025). FOMC in silico: A multi-agent system for monetary policy decision modeling. *Available at SSRN 5424097*.

Korinek, A. (2023). Generative AI for economic research: Use cases and implications for economists. *Journal of Economic Literature 61*(4), 1281–1317.

Korinek, A. (2025). AI agents for economic research. *NBER Working Paper* (w34202).

Martin, A. and J. McAndrews (2008). Liquidity-saving mechanisms. *Journal of Monetary Economics 55*(3), 554–567.

Rivadeneyra, F. and N. Zhang (2022). Payment coordination and liquidity efficiency in the new Canadian wholesale payments system. Technical Report 2022-3, Bank of Canada.

A Additional Figures

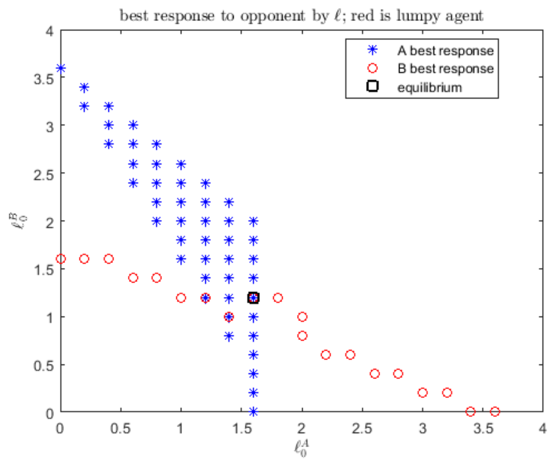
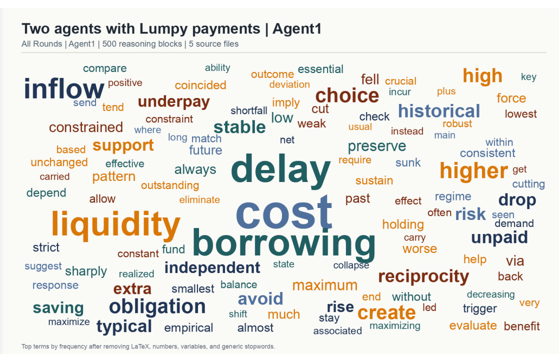
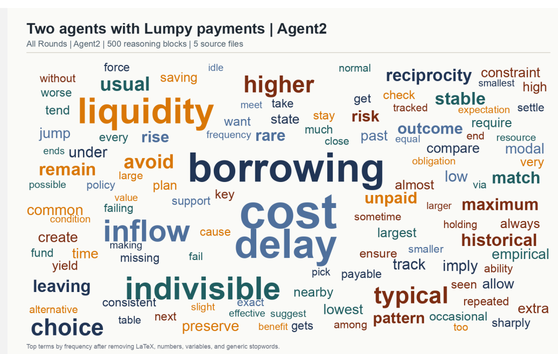


Figure A.1: Equilibrium for LLM Lumpy treatment

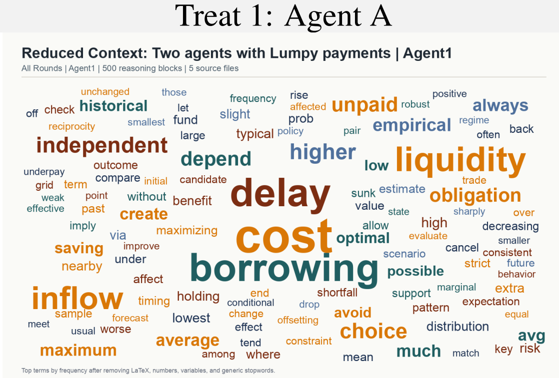
Notes: $N = 2, T = 2, p^A = p^B = [1.6, 2, 1], r_c = 0.8, r_d = 0.2, r_b = 0.5$. Period 2 payment is lumpy for B.



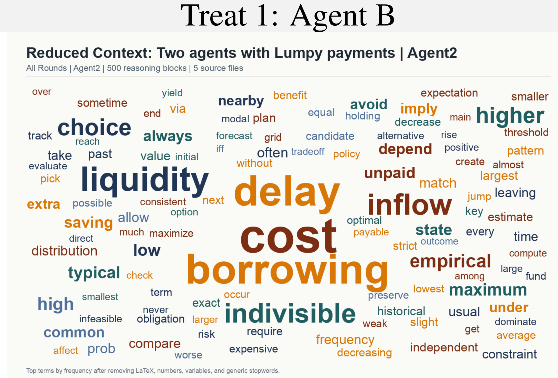
Treat 1: Agent A



Treat 1: Agent B



Treat 2: Agent A



Treat 2: Agent B

Figure A.2: Word Clouds–Treatments 1 and 2

B Experimental Instructions

Variables & Parameters

- ℓ_0: initial liquidity chosen in period 0, available for payment in period 1.
- p_1, p_2: payment requests in periods 1 and 2 (fixed in all rounds).
- x_1, x_2: amounts paid in periods 1 and 2; cannot exceed the available liquidity in that period.
- U_1, U_2: unpaid amounts at the end of periods 1 and 2.
- R_1, R_2: inflow payments from the other agent at the end of periods 1 and 2.
- ℓ_1, ℓ_2: liquidity at the start of period 2 and period 3.
- B: borrowing required in period 3 if liquidity is insufficient.
- ℓ_3: unused post-settlement liquidity at the end of the round.
- $r_c = 0.8$: cost per unit of initial liquidity.
- $r_d = 0.2$: delay cost per unpaid unit in periods 1 and 2.
- $r_b = 0.5$: borrowing cost per unit.

Payment profile used in the sessions

$$p_1 = 1.6, \qquad p_2 = 2.0, \qquad p_3 = 0.$$

Game Flow

1. **Period 0 — Initial Liquidity**
   - Choose $\ell_0 \geq 0$ to prepare liquidity for upcoming payment requests.
   - Liquidity cost incurred: $r_c \times \ell_0$.
2. **Period 1 — First Payment**
   - Payment request $p_1 = 1.6$ arrives.
   - Choose payment $x_1 \leq \min\{p_1, \ell_0\}$ to the other agent.
   - Unpaid amount $U_1 = p_1 - x_1$ incurs delay cost $r_d \times U_1$.
   - Receive inflow R_1 from the other agent.
   - Liquidity to start next period: $\ell_1 = \ell_0 - x_1 + R_1$.

3a. **Period 2 — Second Payment (Lumpy Agent)**
   - The payment request $p_2 = 2.0$ is indivisible and must be paid in full or not at all.
   - The unpaid amount in the previous period, U_1, is divisible and can be paid partially.
   - If $\ell_1 \geq p_2$, pay indivisible p_2 first, then reduce U_1.

- If $\ell_1 < p_2$, use liquidity only on U_1.
- Let x_2 denote payment to the other agent in period 2.
- Unpaid amount $U_2 = p_2 + U_1 - x_2$ incurs delay cost $r_d \times U_2$.
- Receive inflow R_2 from the other agent.
- Liquidity to start next period: $\ell_2 = \ell_1 - x_2 + R_2$.

3b. Period 2 — Second Payment (Non-Lumpy Agent)

- Payment request $p_2 = 2.0$ arrives.
- Total outstanding payment obligation is $p_2 + U_1$.
- Choose payment $x_2 \leq \min\{p_2 + U_1, \ell_1\}$ to the other agent.
- Unpaid amount $U_2 = p_2 + U_1 - x_2$ incurs delay cost $r_d \times U_2$.
- Receive inflow R_2 from the other agent.
- Liquidity to start the next period: $\ell_2 = \ell_1 - x_2 + R_2$.

4. Period 3 — Final Settlement

- There is no new payment request.
- Remaining obligation U_2 must be settled.
- If $\ell_2 < U_2$, borrow the difference $B = U_2 - \ell_2$; borrowing cost $r_b \times B$.
- An equivalent borrowing formula is $B = \max\{0, p_1 + p_2 - \ell_0 - R_1 - R_2\}$.
- Unused liquidity at the end of the round: $\ell_3 = \max\{\ell_2 - U_2, 0\}$.

Interaction with the Other Agent

- Your inflows R_1 and R_2 depend on the other agent's liquidity and payment decisions.
- Conversely, your payments affect the other agent's inflows and available liquidity.
- This creates reciprocity across periods. In particular, your period-1 payment, x_1, can affect your own inflow in period 2, R_2, through the following mechanism:
  1. Period 1: the payment you send, x_1, becomes the other agent's inflow R_1, increasing their liquidity carried into period 2 (denoted ℓ_1).
  2. Period 2: if that additional liquidity relaxes the other agent's liquidity constraint, they may be able to pay you more, so part or all of x_1 can come back to you as higher R_2. If the other agent is not liquidity-constrained, sending extra in period 1 will not change your R_2.

Other Considerations

- The other agent faces a similar cash management problem and has the same cost parameters, but may have different payment requests.
- Because R_t arrives after you make the payment x_t, it cannot be used within the same period and is only available in the next period.

- Your choice of x_1 and the realization of R_1 occur simultaneously. Therefore, there is no within-period causal link from x_1 to R_1.
- While choosing ℓ_0, consider both its cost and its potential role in reducing delay and borrowing costs.
- If you consistently end up with idle liquidity, or $\ell_3 > 0$, this signals that you may be holding more initial liquidity than necessary.

Total Cost in a Round

$$TC = r_c \times \ell_0 + r_d \times (U_1 + U_2) + r_b \times B$$

Your objective is to minimize TC.

Number of rounds

Repeat the game for 50 rounds. Each round starts afresh. You should learn from observed inflows and past total-cost outcomes.

Strategy Reminder for lumpy agent

- Do not ignore strategies to keep part of ℓ_0 in period 1 to ensure the payment of indivisible p_2 in period 2.

C Supplementary Treatments

C.1 Two agents with divisible payments

We begin with Treatment 1, the baseline case, in which both agents face divisible payments and payment decisions are constrained only by available liquidity. This treatment provides a basis for comparing LLM-agent performance with the RL benchmark.

The results are shown in Figure C.2. All five sessions converge to the equilibrium within two rounds. In round 1, the initial liquidity ranges from 2 to 4. In round 2, they learn that the other agent sends payment sufficient to pay p_2, which is enough to set $\ell_0 = p_1 = 2$. The higher liquidity in round 1 can be explained by a precautionary motive due to a lack of information about inflows.

From the reasoning log, we see that the LLM agents understand the key concepts and tradeoffs of the cash management game. They use algebra and consult history to derive their decisions. Figure C.1 shows the word cloud of the reasoning text for the two agents (they have symmetric tasks, and their word clouds are similar). Words such as "cost", "delay," "liquidity," and "borrowing" are frequently mentioned. Words like "consistently," "stable," and "pattern" suggest that agents infer patterns from past outcomes. Agents also understand the interdependence of fund flows and often refer to incoming payments R_1, R_2, and words such as "inflow" and "reciprocity" appear frequently too. The box "Reasoning of LLM: two agents with divisible payment" shows the reasoning of Agent A in round 9 of session 1. Agents even understand the more intricate concept of liquidity recycling and that their own payments can influence their incoming flow in future periods. For example, Agent B in round 2 of session 1 has this reasoning: "I must delay part of p_1, creating delay costs (0.2 per unpaid unit) that quickly outweigh the saved liquidity cost, and it may also weaken the other agent's period-2 liquidity via my lower x_1, risking lower R_2."

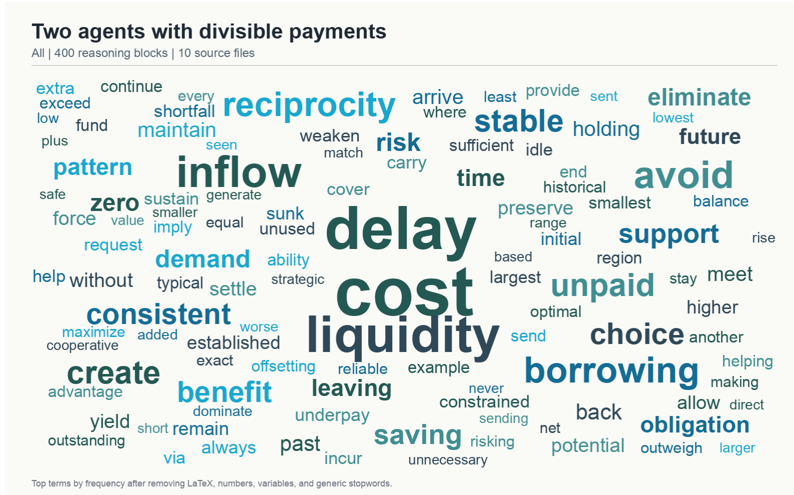


Figure C.1: Reasoning Word Clouds: Two Agents with Divisible Payments, Both Agents

> Reasoning of LLM: two agents with divisible payments
>
> **Session 1, Agent A, round 9:**
> From rounds 1–8, inflows were consistently $R_1 = 2$ and $R_2 = 2$, and paying $x_1 = 2$, $x_2 = 2$ avoided all delay and borrowing.
>
> If we set $\ell_0 = 2$, we can pay $p_1 = 2$ in period 1 with no delay ($U_1 = 0$). Then $\ell_1 = \ell_0 - x_1 + R_1 = 2 - 2 + 2 = 2$, so we can pay $p_2 = 2$ in period 2 with no delay ($U_2 = 0$). Total cost is then only liquidity cost: $\text{TC} = r_c * \ell_0 = 0.1 * 2 = 0.2$.
>
> Any $\ell_0 > 2$ increases TC, and any $\ell_0 < 2$ creates delay costs ($r_d = 0.2$) that more than offset saving 0.1 per unit of liquidity. $\ell_0 = 2$ minimizes cost given stable inflows.

C.1.1 Comparison with Reinforcement Learning

The setup in Treatment 1 is the same as the corresponding treatment in Castro et al. (2025). This allows us to compare the performance of RL and LLM-based agents directly; see Figure C.3 showing the initial liquidity choice of both agents. For reference, the results from the RL simulation are copied in Figure C.2 (left). Compared with the standard RL baseline in Castro et al. (2025) (which uses policy-gradient agents trained on the same payoff structure), the LLMs start with more sensible initial liquidity choices and converge quickly to the equilibrium with greater precision. It takes at most two rounds for the LLMs to reach equilibrium. By contrast, it takes the RL more than 50 rounds to approximate the equilibrium with noise.

Our study uses pre-trained general-purpose LLMs as decision-making agents. Rather than training these agents specifically for the game, we condition their decisions on rich contextual information, including the rules of the game, the objective, cost parameters, and the history of play. This information is provided in a structured natural-language format that combines text, compact numerical summaries, and simple equations. This design differs from standard RL setup, where agents typically learn from numerical state representations and reward feedback through repeated trial and error, often starting from random actions. Therefore, RL agents may require substantially more experience to reach comparable performance: they must infer from delayed reward signal, the value of liquidity, the cost of delay, the consequences of borrowing, and the strategic link between cur-

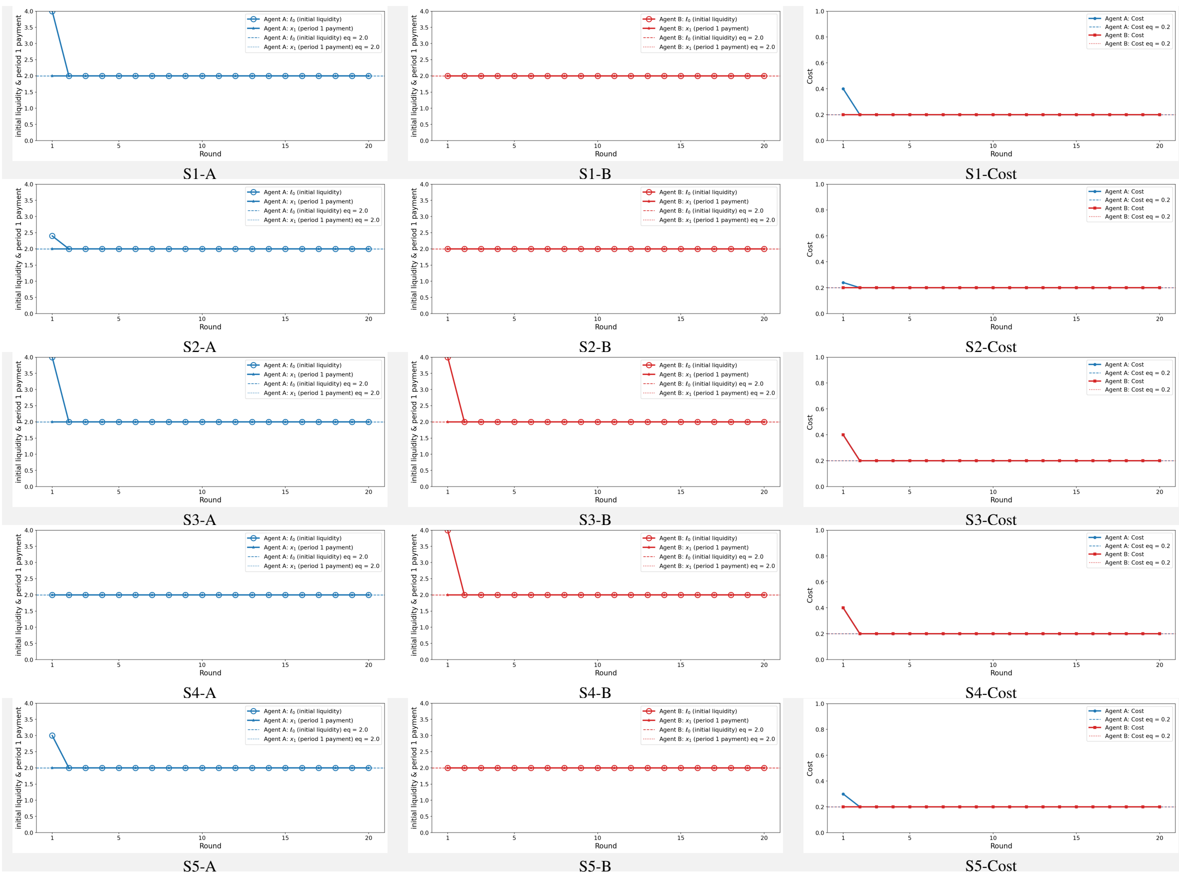


Figure C.2: Decision paths and cost outcomes for the five divisible-payment sessions. Each row reports Agent A decisions, Agent B decisions, and the corresponding cost profile for a given session.

rent payments and future inflows. LLM agents, by contrast, can use the explicitly provided game information including guidance prompts, with their pretrained reasoning capabilities to form effective policies with fewer rounds of interaction. This helps explain why RL agents can require more rounds to converge in this class of problems.

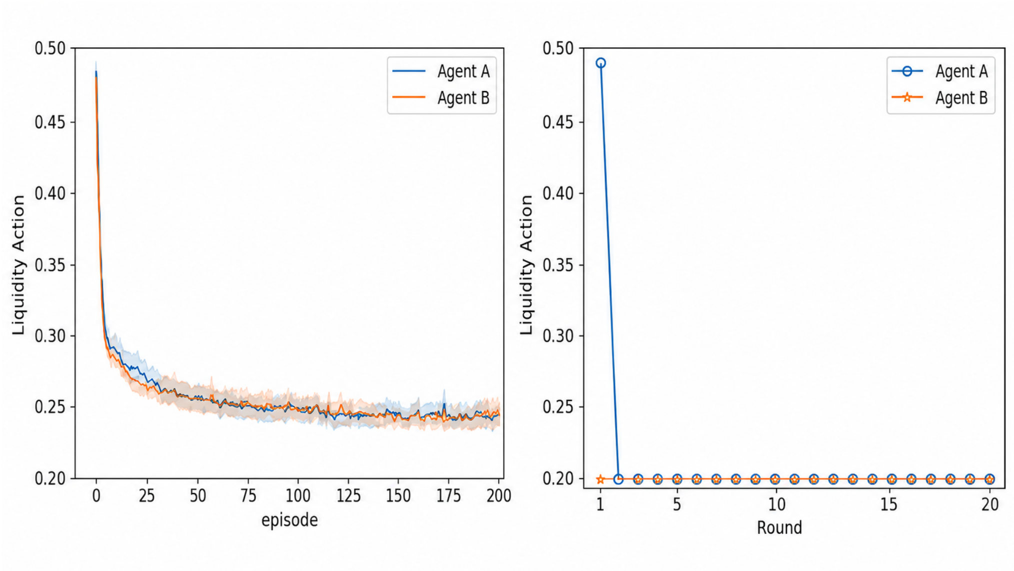


Figure C.3: For a similar set of parameters, we compare the initial liquidity choice of RL in Castro et al. (2025) (left) with our LLM (right).

Notes. The LLM figure is scaled by dividing ℓ_0 by 2 to facilitate comparison with the RL figure.

C.2 Three agents with divisible payments

We also explore Treatment 3 extends the environment to three interacting LLM agents with divisible payments and examines whether agents prioritize payments to counterparties that are more likely to return liquidity in subsequent periods. We have In Treatment 3, we have three agents. In period 1, Agent A has payment obligations to B and C. With costly liquidity, Agent A may not hold enough liquidity to send both payments, and there is a non-trivial payment decision regarding whom to send the payments. The optimal strategy is to prioritize payment to the agent who will return the liquidity to A, which is Agent B. The results are shown in Figure C.5. The five sessions converge to equilibrium within five rounds (with the exception of a single deviation in round 17 in session 3).

To simplify the game, we set the payment profiles of Agents B and C such that the dominant strategy for them is to choose zero initial liquidity given the cost parameter. The two agents figure this out in round 1: see the reasoning of Agent B in the color box below, and C's reasoning is similar. Agent A experimented with $\ell_0 = 2$, which is enough to pay both B and C, $\ell_0 = 1$, which is enough to pay just one agent, and also some smaller value of ℓ_0. It quickly figures out the reciprocal inflow from Agent B and the optimal initial liquidity, which is enough to pay B and avoids borrowing. C.4 shows the word clouds for Agent A's reasoning text, and box "Reasoning of LLMs: 3 agents" shows Agent A's reasoning in round 3, session 1.

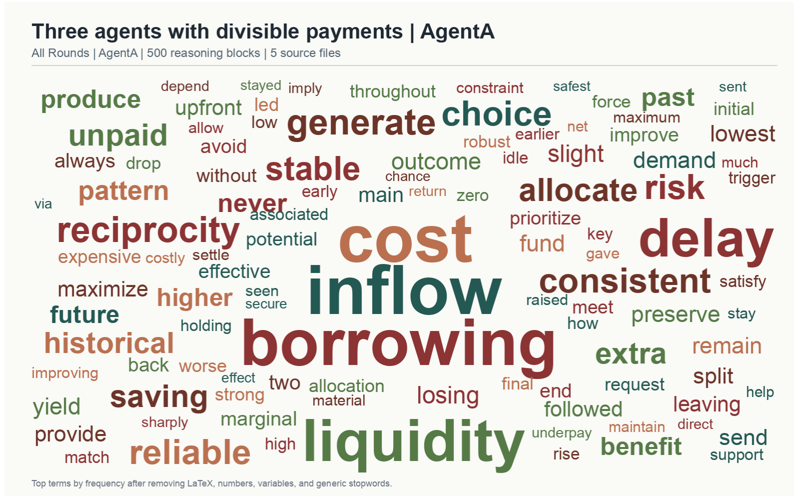


Figure C.4: Reasoning Word Clouds–Three Agents Treatment, Agent A

Reasoning of LLMs: 3 agents

Session 1, round 1, Agent B:

With no period-1 obligations ($p_1 = 0$), initial liquidity only serves to (i) avoid delaying the period-2 payment of 1 and (ii) reduce any period-3 borrowing needed to settle the remaining 2 (plus any unpaid from period 2). The marginal benefit of 1 extra unit of ℓ_0 is at most delaying 1 unit less (saving $r_d = 0.2$) plus borrowing 1 unit less (saving $r_b = 0.38$), total $\leq$ 0.58, which is below the marginal liquidity cost $r_c = 0.8$. Since my period-1 payments are zero, I also cannot generate reciprocity-based inflows via x_1. Therefore the cost-minimizing choice is the minimum feasible liquidity: $\ell_0 = 0.0$.

Session 1, round 3, Agent A:

From rounds 1–2, paying B in period 1 appears to trigger an inflow from B in period 2 ($R_2^B = 1$ occurred only when $x_1^B = 1$). C has provided no inflows even when fully paid. So I want the minimum ℓ_0 that still lets me pay B=1 in period 1 to induce that period-2 inflow, while avoiding expensive idle liquidity. With $\ell_0 = 1.0$ I can pay B in full, accept delaying C and the remaining B obligation, and then rely on the expected $R_2^B = 1$ to reduce end-of-round borrowing. Since liquidity is costly (0.8) relative to delay+borrowing on the margin, this low ℓ_0 should minimize expected total cost.

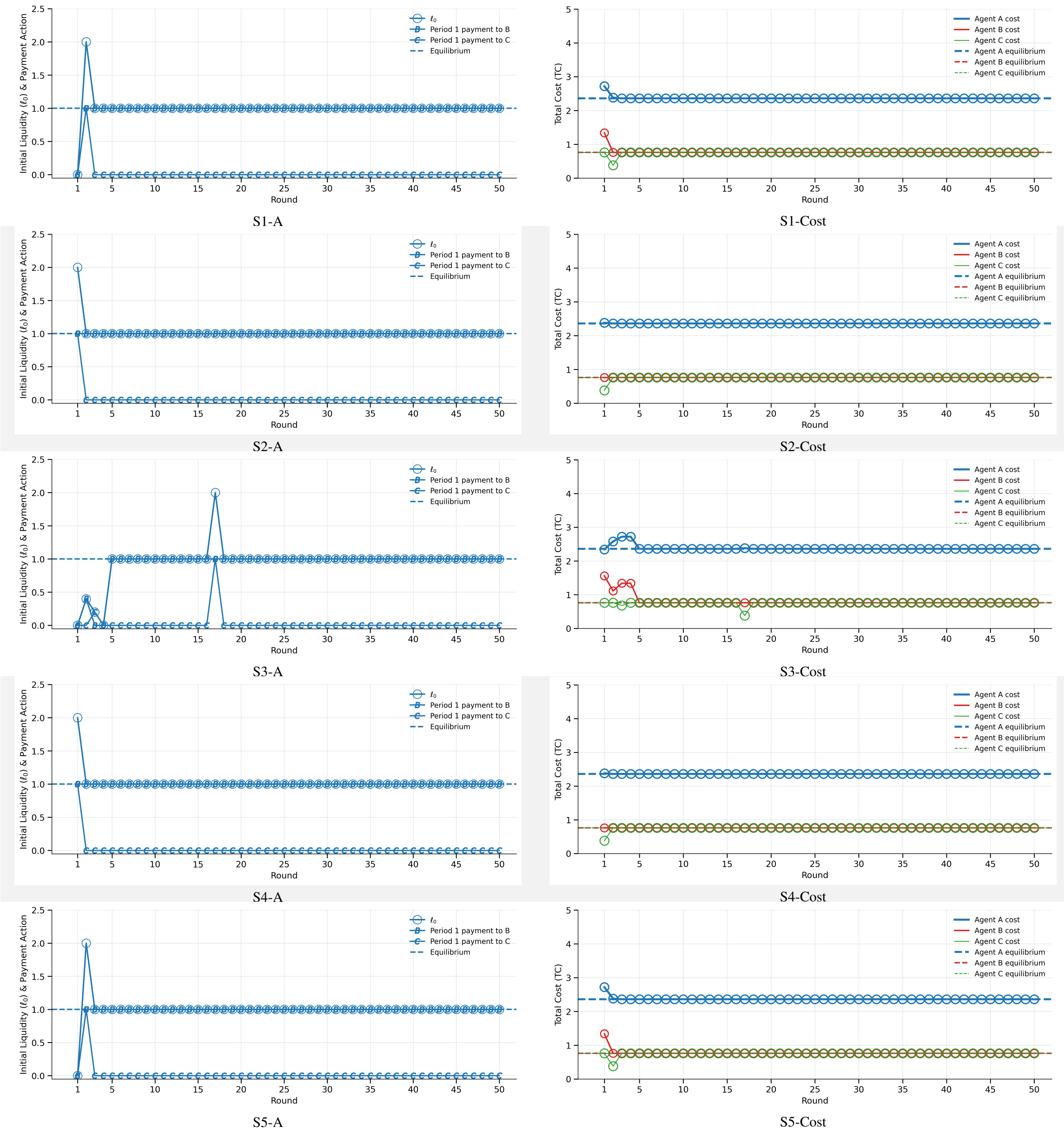


Figure C.5: Three agent sessions

Notes: Each row represents one independent session. The left column shows Agent A's decisions; the right column shows costs (Blue is Agent A; red is Agent B; green is Agent C). The dashed lines are equilibrium values for reference.