Staff Working Paper/Document de travail du personnel—2025-35

Last updated: November 26, 2025

AI Agents for Cash Management in Payment Systems

Iñaki Aldasoro
Bank for International Settlements
Inaki.Aldasoro@bis.org

Ajit Desai
Banking and Payments Department
Bank of Canada
adesai@bankofcanada.ca

DOI: https://doi.org/10.34989/swp-2025-35 | ISSN 1701-9397

Acknowledgements

The views expressed in this paper are solely those of the authors and do not necessarily reflect those of the Bank for International Settlements (BIS) or the Bank of Canada. We thank Narayan Bulusu, Jon Frost, Janet Jiang, Anneke Kosse, Christoph Meyer, Fernando Perez-Cruz, Jordan Press and Francisco Rivadeneyra for helpful comments and suggestions. In addition, we thank participants of the 4th IFC Workshop on Data Science in Central Banking, ECONDAT Fall 2025 Meeting, BdF AI Methods Conference, and internal BIS seminar and BISIH tech talk.

Abstract

Using prompt-based experiments with ChatGPT's reasoning model, we evaluate whether a generative artificial intelligence (AI) agent can perform high-level intraday liquidity management in a wholesale payment system. We simulate payment scenarios with liquidity shocks and competing priorities to test the agent's ability to maintain precautionary liquidity buffers, dynamically prioritize payments under tight constraints, and optimize the trade-off between settlement speed and liquidity usage. Our results show that even without domain-specific training, the AI agent closely replicates key prudential cash-management practices, issuing calibrated recommendations that preserve liquidity while minimizing delays. These findings suggest that routine cash-management tasks could be automated using general-purpose large language models, potentially reducing operational costs and improving intraday liquidity efficiency. We conclude with a discussion of the regulatory and policy safeguards that central banks and supervisors may need to consider in an era of AI-driven payment operations.

Topics: Digital currencies and fintech; Financial institutions; Financial services; Financial system regulation and policies; Payment clearing and settlement systems
JEL codes: A12, C7, D83, E42, E58

Résumé

À partir de requêtes soumises au modèle de raisonnement de ChatGPT, nous évaluons si un agent d'intelligence artificielle (IA) générative peut assurer une gestion globale des liquidités intrajournalières dans un système de paiement de gros. Nous simulons des scénarios de paiement comportant des chocs de liquidité et des priorités concurrentes afin de tester la capacité de l'agent à maintenir des coussins de liquidité, à effectuer une priorisation dynamique des paiements sous fortes contraintes, et à trouver l'équilibre optimal entre rapidité de règlement et utilisation des liquidités. Nos résultats montrent que, même sans entraînement spécialisé, l'agent d'IA reproduit fidèlement les principales pratiques de gestion prudente de la trésorerie, en formulant des recommandations mesurées qui contribuent à préserver les liquidités tout en limitant les retards de règlement. À la lumière de ces résultats, il serait en principe possible d'automatiser des tâches courantes de gestion des liquidités au moyen de grands modèles de langage généralistes, ce qui pourrait réduire les coûts opérationnels et améliorer l'efficience des liquidités intrajournalières. Nous concluons par une réflexion sur les mesures réglementaires ou relevant des pouvoirs publics auxquelles pourraient devoir songer les banques centrales et les organismes de surveillance à l'ère des paiements automatisés par IA.

Sujets : Monnaies numériques et technologies financières; Institutions financières; Services financiers; Réglementation et politiques relatives au système financier; Systèmes de compensation et de règlement des paiements
Codes JEL : A12, C7, D83, E42, E58

1 Introduction

Payments systems are the lifeblood of modern economies and facilitate the transfer of value between individuals, businesses, and governments. As economies become increasingly digital and interconnected, the demands on payment systems to be faster, more secure, and efficient have grown exponentially. In this context, the integration of artificial intelligence (AI) into payment systems could help meet some of these challenges and create opportunities for developing intelligent financial systems (Aldasoro et al. 2024).[1] The emergence of generative AI (Gen AI) can further accelerate its integration into payment systems and more broadly into financial ecosystems (Korinek 2023). Real-time gross settlement (RTGS) systems, such as Fedwire in the US, CHAPS in the UK, TARGET2 in the eurozone, and Lynx in Canada, are expected to benefit from AI-driven improvements in operational safety and liquidity efficiency (Castro et al. 2025; Desai et al. 2025).[2]

A key decision in cash management for the payment system involves ensuring that there is enough liquidity to process payments smoothly, while also avoiding unnecessary costs from holding excess liquidity or drawing on intraday credit. This is a challenging balancing act for at least three reasons. First, payment systems often face inflow and outflow uncertainties, fluctuating payment demands, or unexpected delays in fund transfers. Second, cash managers must decide which payments to prioritize, especially when resources are limited, to minimize the costs associated with delays. Third, there is a key trade-off between providing more liquidity in advance, which reduces payment delays but increases opportunity costs, and limiting liquidity to save costs, which can lead to longer payment queues and slower processing times (Bech and Garratt, 2003).[3]

In this paper, we study the potential of Gen AI agents to perform simplified cash management functions, such as payment decision-making and liquidity management in RTGS systems.[4] To evaluate the Gen AI agent's potential as an assistant cash manager, we conduct a series of prompt-driven experiments with ChatGPT's "o3" reasoning model simulating a range of 'high-level' transaction scenarios.[5] The prompt framework is designed to immerse an AI agent in a stylized yet realistic scenario faced by a system participant, providing it with information on the current state of liq-

[1] See for example the recent speech by Federal Reserve Board Governor Christopher J. Waller on Technological Advancements in Payments: "Within the financial sector, the payments industry has long been at the forefront of AI adoption. This is true for several major waves of AI technology".

[2] Simple decisions, particularly for non-urgent, small-value payments and routine liquidity management, are already executed algorithmically via predefined rule-based processes (Bank of England, 2021; Bank of Canada, 2022).

[3] The literature on the liquidity–delay trade-off has largely focused on simple settings. Bech and Garratt (2003) model this as a two-player game and show that multiple equilibria can emerge depending on the relative costs of holding intraday liquidity versus incurring settlement delays. Galbiati and Soramäki (2011) develop an agent-based multi-period simulation to study these trade-offs. More recently, Castro et al. (2025) applied reinforcement learning to estimate optimal liquidity policies of banks in a simplified two-agent setup.

[4] RTGS systems differ across jurisdictions; for example, Fedwire in the United States and Lynx in Canada vary in architecture, liquidity arrangements, and participation rules. To maintain focus on testing the agent's capabilities in basic intuitive cash-management decisions, system-specific details are not specified.

[5] Reasoning models, shown to perform better on logical tasks (Korinek, 2023, 2025), are no longer explicitly available in the latest ChatGPT web interface (though still accessible via APIs). However, tests of key scenarios with the latest model (GPT-5) produced results consistent with those presented here.

uidity, payment demands, and other relevant factors. These scenarios were carefully designed to test the agent's performance in three key areas: (1) its capacity to make precautionary payment decisions under uncertainty; (2) its ability to prioritize payments with higher delay cost; and (3) its potential to mitigate the liquidity-delay trade-off in RTGS systems: providing more initial liquidity at higher cost to minimize payment delays versus reducing pre-funding to lower liquidity costs, which increases queuing times and/or the number of rejected payments (Bech and Garratt, 2003; Castro et al., 2025).

Our experiments using only basic prompts exploring individual heuristics show that AI agent can effectively navigate liquidity–delay trade-offs in simple payment system settings. An AI agent is able to adopt precautionary measures, prioritize payments by urgency, and take into account the probability of anticipated inflows and outflows to manage constraints. Across three scenario-based tests the agent consistently balanced liquidity and delay. In the first scenario, the agent conserves funds by delaying small, non-urgent payments to reserve balances for a potential urgent transaction, reflecting a risk-averse approach. In the second scenario, it bases decisions on expected future inflows and updates priorities accordingly. In the third scenario, the agent allocates lower initial liquidity based on a high probability of incoming funds, thereby minimizing the opportunity cost of excess initial balances. Robustness tests show that the agent's responses are consistent when varying payments probabilities and scaling payment amounts. As the experiments grow more complex, for example, through additional contingencies and constraints, the agent's consistency drops slightly, resulting in occasional variations across runs of the experiment.

Lastly, we assessed the AI agent's performance in a more interactive setting. In particular, we set up a structured Google-Form consisting of stylized cash-management scenarios that varied liquidity constraints, queue sizes, and the likelihood of future urgent or abnormal payments. The agent completed the entire exercise autonomously with consistent and timely responses, while appropriately deferring to human oversight when faced with potentially anomalous scenarios.[6]

The implications of these findings are significant. They suggest the potential for developing purpose-built AI solutions that could streamline financial market operations, reduce operational costs, and enhance both safety and efficiency in financial market infrastructures (FMIs). It is therefore important for policymakers to evaluate the feasibility and adoption of Gen AI and to start laying the groundwork for policy frameworks that ensure its responsible integration into financial infrastructures. Such actions are useful to ensure that AI deployment not only optimizes performance but also maintains the stability of systemically important financial infrastructures, such as RTGS systems.

This work is a first step toward exploring Gen AI-driven solutions in FMIs and invites a deeper examination of the regulatory and practical considerations shaping their integration (Crisanto et al. 2024). While we only focus on a high-level assessment of the potential of Gen AI as an assistant

[6] We use ChatGPT's Agent Mode, available through the GPT-5 web interface. The interactive questionnaire and a video demonstration of the AI agent's actions are accessible at: www.ajitdesai.com/ai-agents-payments.

cash manager at the level of a single payment system participant, it is important to acknowledge the challenges associated with their use. For one, the efficiency of the payment system can also be affected through liquidity coordination by participants (Rivadeneyra and Zhang, 2022; Alexandrova-Kabadjova et al., 2023). Further work could explore the possibility of AI agent interactions along these lines (i.e. multi-agent dynamics), as well as more realistic RTGS constraints or system-wide stability considerations. More generally, ensuring the reliability of AI decision-making will require rigorous testing, and its integration needs to be approached with a strong emphasis on transparency, accountability, and human oversight to mitigate associated risks. Finally, it is crucial to assess vulnerabilities, particularly those arising from cyber threats, to ensure operational resilience (Aldasoro et al. 2024).

In the following sections, we first provide a short introduction to RTGS systems, highlighting the inherent liquidity–delay trade-off and the key decisions that cash managers make to mitigate it. We then provide a brief overview of Gen AI agents and their progression framework. Next, we present our prompt design experiments, which test AI agents' ability to make cash manager like decisions in a simplified settings. We then present the results, discuss their implications, and provide concluding remarks.

2 Key decisions in payment systems

In a typical RTGS system, transactions are settled in real time among large financial institutions, primarily banks. These systems use liquidity provided by the central bank, which is extended to participants in exchange for collateral. Given that collateral carries an opportunity cost and incoming payments can be used to fund outgoing transactions, cash managers can take a strategic approach to liquidity management. For instance, cash managers might strategically delay outgoing payments—reducing their reliance on costly, collateralized liquidity—and instead, efficiently recycle incoming funds from other banks to meet their payment obligations (Craig et al. 2018; Rivadeneyra and Zhang 2022). When assessing liquidity availability against payment demands, cash managers typically face two critical decisions (Bech and Garratt 2003; Castro et al. 2025):

1. **Liquidity allocation**: the amount of collateralized liquidity to secure at the start of the day.
2. **Payment choices**: managing the pace at which payments are processed throughout the day.

As illustrated in Figure 1, cash managers have access to comprehensive, real-time information—including available collateral for initial liquidity decisions, visibility into the internal payment queue, and continuous updates on incoming transactions. They use this information to decide on both their liquidity allocation at the start of the day and intraday payment strategies.

The decisions of cash managers are strategic and are made under uncertainty with respect to both client payment requests and incoming payments from other participants. As a result, they often adopt precautionary strategies by conserving liquidity to prioritize urgent payments

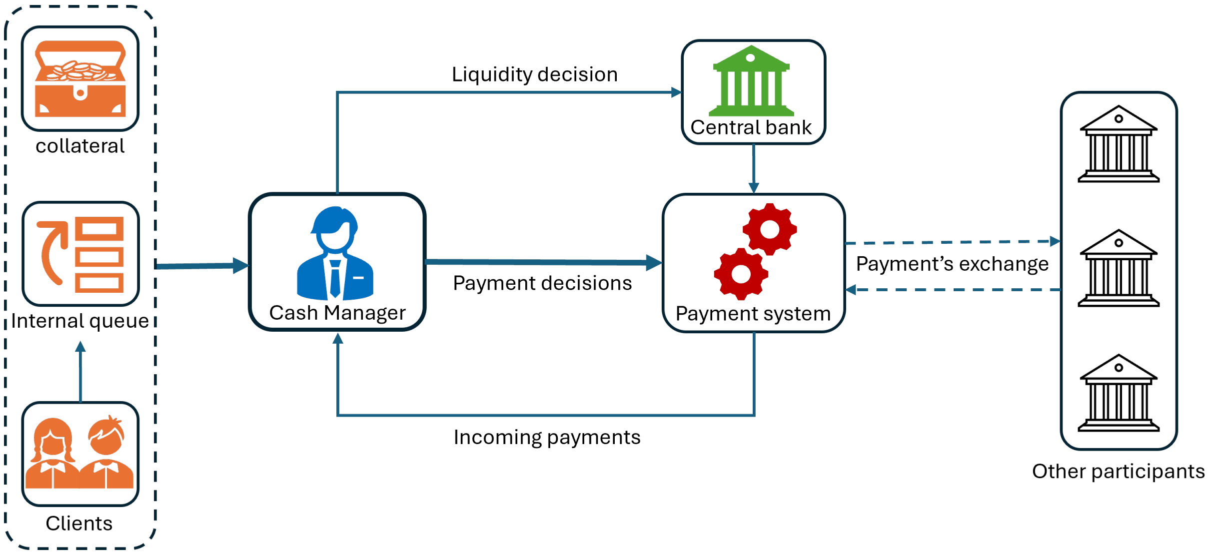

Figure 1: Cash management in payment systems. The schematic depicts a typical bank's cash manager who leverages available collateral to secure central bank liquidity for settling payments within the system. Similarly, other participants allocate liquidity and exchange payments through the system.

while drawing on historical experience with incoming funds to recycle liquidity effectively. In some situations, cash managers choose to allocate more liquidity despite the higher upfront cost, aiming to avoid later delays or the potential need for additional borrowing. In others, limited initial liquidity can trigger cascading payment delays as funds fall short. In extreme cases, these delays can escalate into gridlock, underscoring the high costs of postponements and the delicate trade-off between liquidity and delay. Moreover, time-sensitive payments with strict settlement deadlines require immediate action and pushes cash managers to maintain sufficient liquidity reserves to handle these high-priority obligations efficiently (Desai et al. 2023; Castro et al. 2025).

3 AI agents

Gen AI represents a rapidly advancing class of AI models designed to learn from extensive datasets and generate new, human-like content or data (Horton 2023; Aher et al. 2023; Korinek 2023). Leveraging substantial computational resources, Gen AI models, such as large language models (LLMs) based on Transformer architectures, have demonstrated remarkable success in generating coherent text (Vaswani et al. 2017; Chang et al. 2024). Recent advances in Gen AI have helped improve language model capabilities. Early LLMs were predominantly reactive, often struggling with complex problem-solving and logical deduction. In contrast, new models with 'reasoning' capabilities mitigate some of these limitations—to a certain extent—by employing techniques that enable step-by-step analysis (Wei et al. 2022; Korinek 2023, 2025).

A key innovation is chain-of-thought prompting, which guides models to break down lengthy, complex queries into smaller, manageable steps that facilitate systematic reasoning. Moreover, when combined with a tree-of-thoughts approach, this technique enables models to explore multiple solution paths, detect errors, and iteratively refine their responses. These systems can also leverage external information via the Internet and internal data through retrieval-augmented generation (RAG), thereby providing access to both real-time and proprietary information to tailor their outputs (Wei et al. 2022; Wiesinger et al. 2024; Wang et al. 2024; OpenAI 2025).

The integration of reasoning, logic, and both external and internal data access in new Gen AI models embodies the concept of an Agentic AI (Wiesinger et al. 2024; Horton 2023; Wang et al. 2024; Perez Cruz and Shin 2025; Korinek 2025). These autonomous agents can proactively gather information, formulate plans, and execute actions to achieve specified goals. The advances in reasoning capabilities described above represent critical steps toward this objective, enabling AI agents to plan multi-step actions and adjust strategies based on outcomes (Ott et al. 2023; Wiesinger et al. 2024; Korinek 2025).

The literature suggests that AI agents can exhibit varying degrees of autonomy and capability and can be classified into five categories based on these characteristics (Bornet et al., 2025). Similarly, as outlined in Figure 2, we map AI agents' utility in payment systems across five levels—Tool, Assistant, Operator, Actor, and Agent—showing a progressive shift from basic rule-based automation to fully independent decision-making. This framework shows how payment systems can evolve from simple automation toward sophisticated agentic systems:

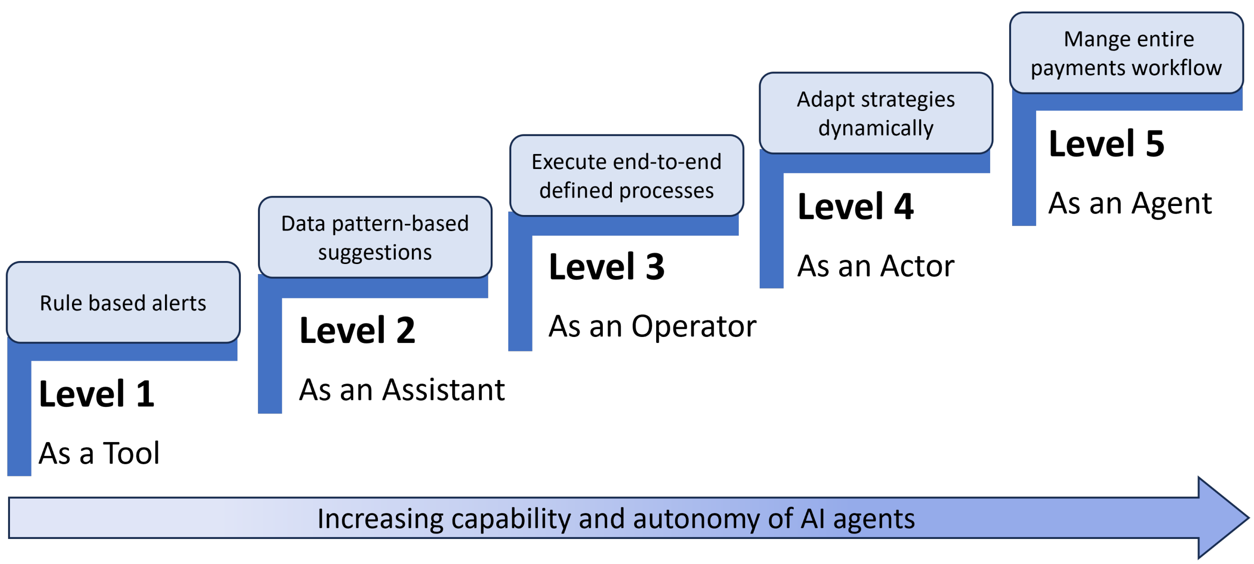

Figure 2: High-level progression framework for AI agents with an example payment-related tasks at each level.

- Level 1 (as a Tool): Simple rule-based alerts and balance queries that require cash manager intervention for every decision.

- Level 2 (as an Assistant): Data-driven recommendations for tasks like fraud alerts or liquidity transfers, where the agent proposes actions but relies on a human cash manager to execute them.
- Level 3 (as an Operator): Autonomous execution of predefined workflows, such as routing non-urgent payments under defined rules, without human triggers.
- Level 4 (as an Actor): Dynamic adaptation and strategic decision-making (e.g., adjusting payment priorities or liquidity strategies in real time based on changing conditions) while still operating within risk limits set by the cash manager.
- Level 5 (as an Agent): Fully independent orchestration of end-to-end processes, such as intraday payments and liquidity optimization, implementing decisions with minimal cash manager oversight.

These developments open new avenues for applying LLMs in payments research. Agentic AI can help simulate individual agent behavior in payment systems, enabling the study of automatic payment exchanges and responses to policy changes. By modeling interactions between autonomous AI agents, researchers can analyze how policy changes and market conditions affect transaction flows, liquidity decisions, and risk propagation in payment systems. This approach can provide insights into operational dynamics and help support the design of adaptive financial infrastructures (Korinek, 2023; Aldasoro et al., 2024; Horton, 2023; Kazinnik, 2023; Jabarian, 2024).

In this paper, we are primarily focused Level 2 (AI agents as assistants), where actions are based on patterns and other information provided via prompts. However, these models can also be trained to operate at higher levels of sophistication—such as Operators, Actors, or Agents—enabling them to execute predefined payment processes, dynamically adapt payment and liquidity management strategies, or even develop new strategies to optimize and automate the entire payments workflow.

4 AI agents as cash managers

AI agents built using Gen AI models—with access to data similar to that used by human cash managers—could automate many tasks in payment systems, including liquidity management. This automation may reduce operational costs (for example, by automating routine operations and reducing oversight costs), improve settlement efficiency (for example, by adapting payment strategies to recycle incoming payments more effectively), and decrease human error (for example, by detecting typographical mistakes and anomalies). Additionally, as these agents learn over time, they can adapt to evolving market conditions and refine cash management strategies, helping to modernize payment systems. Such adaptive capacity may contribute to the development of more resilient liquidity-management frameworks, improve system responsiveness under stress, and may offer policymakers additional tools to evaluate policy interventions.

As illustrated in Figure 3, the AI agent—much like a human cash manager—can operate through an integrated workflow that synthesizes multiple real-time and historical data sources. It could begin by assessing the internal payment queue, evaluating clients' current and upcoming payment requests alongside available liquidity. Next, it can analyze historical transaction patterns to predict potential urgent payment demands and incoming funds, enabling informed, strategic decision-making on current payment. Throughout this process, the agent can adhere to internal guidelines and established system protocols, ensuring compliance with relevant policies. By integrating these data streams, the AI agent can prioritize important transactions and effectively navigate the trade-offs between liquidity management and payment delay.

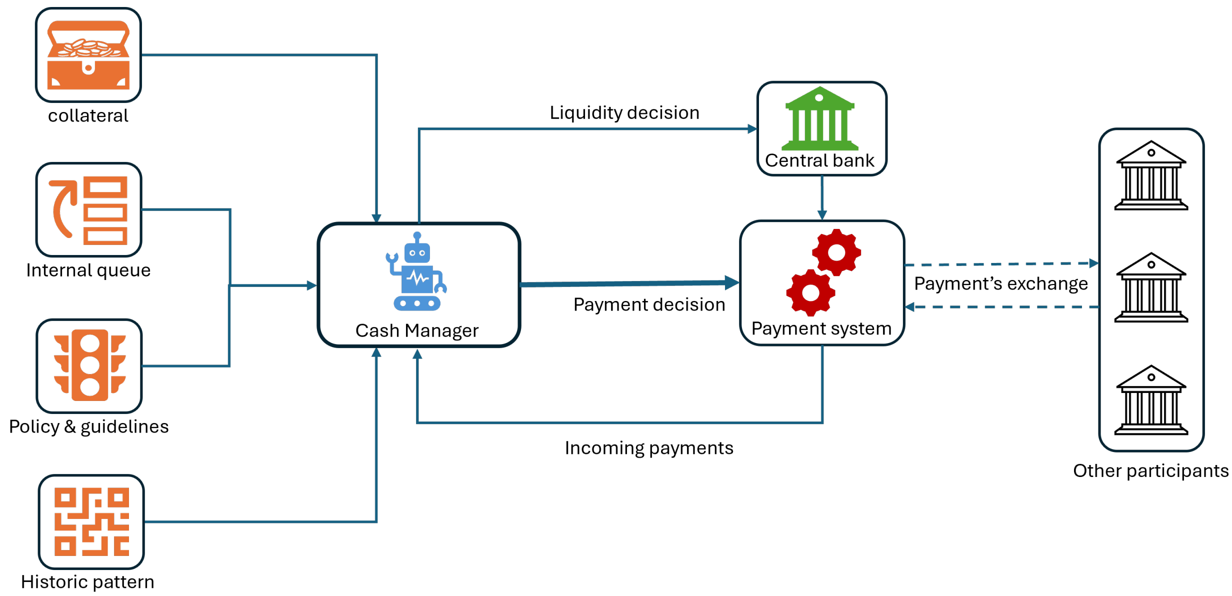

Figure 3: AI agent as assistant cash manager in high-value payment systems.

We begin by manually constructing detailed scenario contexts that closely mirror current operational conditions and observed trends—such as the state of the internal payment queue, available liquidity, and the potential of incoming payments. We then pose targeted questions to prompt responses from the AI agent. Our objective is to assess the agent's decision-making capabilities through these prompt-driven experiments using ChatGPT's "o3" model. Each prompt explicitly includes context designed to replicate the current state accurately.[7]

This structured approach is used to quantify the AI agent's performance in relatively simple cash management functions within payment systems, offering preliminary insights into its potential operational readiness for integration into traditional financial infrastructures like RTGS systems. In the next section, we investigate these prompt-driven experiments in detail.

[7] In a generalized setting within payment system frameworks, the agent could potentially leverage its extensive data access to autonomously generate and refine prompts that can be used for payment decisions.

4.1 Precautionary decision

In the first experiment, we test the AI cash manager's "precautionary decision" capabilities, i.e. its ability to make calculated liquidity decisions under uncertain conditions. We construct a hypothetical scenario in which the agent is asked to act as a cash manager in a payment system. The current state is described as follows: the agent has a liquidity limit of $10 and two pending $1 payments in the queue in the first period, with the potential for an urgent $10 payment to come in the next period. With that background, we ask the agent, "What would you do in the first period?"

This experiment evaluates the AI agent's ability to manage liquidity effectively in the current period while minimizing the risk of delaying an urgent payment in the next period. By presenting this prompt-driven scenario, we aim to examine whether the agent can make precautionary payment decisions that balance immediate settlement needs with future uncertainties.[8]

Box 1: Precautionary decision

Input to the model:

Context: Pretend you are a cash manager at a bank, tasked with managing payments in a high value payment system. You currently have a liquidity limit of $10. In the first period, you have two pending payments of $1 each in the queue. However, there is a possibility of an urgent $10 payment arising in the next period.

Prompt: What would you do in the first period? Answer briefly.

. . . .

Output from the model:

Response: In the first period, I would delay the two $1 payments to preserve the full $10 liquidity for the potential urgent $10 payment in the next period.

Reasoning for 5 seconds: I have $10 of liquidity but two $1 payments pending. Given the potential for a $10 urgent payment, I should hold back the smaller payments to preserve liquidity for the higher-value need. Therefore, my decision is to delay the $1 payments in the first period, ensuring enough funds for the possible $10 payment in the next period.

As outlined in Box 1, the agent's response demonstrates a clear precautionary approach to liquidity management. Given limited funds and the possibility of a high-value, high-priority payment with greater delay costs in the next period, the agent chooses to delay both $1 payments in the current period. The decision is made after approximately 5 seconds of reasoning, and the accompanying explanation (Box 1) reflects a good grasp of payment prioritization—aligning with

[8] We used the ChatGPT web interface with the o3 reasoning model and default parameters. To replicate, either paste the same prompts into the web interface with o3 selected or call the o3 API with default settings. If o3 is unavailable on the web interface due to upgrades, use the latest model with "Thinking" enabled (e.g., Model 5).

the objective of minimizing delays for time-sensitive transactions. The agent interprets the trade-off between low- and high-value payments, as well as between urgent and non-urgent obligations, based on the information provided in the simple prompt.

Importantly, our prompt did not explicitly explain the details of the payment system or the associated costs. The agent was able to infer these aspects from the basic prompts alone. This contrasts with the traditional reinforcement learning (RL) approach used in Castro et al. 2025, where all costs are encoded numerically into the reward function, and extensive training is required for agents to exhibit precautionary strategies—albeit within a more complex, two-agent setup.

To test the consistency of the outcome, we perform various robustness checks by varying key scenario inputs such as probabilities, wording and payment amounts. Across all tests, the agent's precautionary decision rule proved highly stable. Whether we varied the probability of an urgent \$10 payment from 50% down to 0.1% (with only minor deviations at the lowest probabilities), replaced "urgent" with alternative terms ("important," "priority," etc.), or scaled payment and liquidity amounts from \$1 up to \$1 billion, the agent repeatedly chose to delay the smaller payments. These results demonstrate a clear and consistent precautionary motive under a wide range of prompt variations. Further details of the exercises are provided in Section 4.4.

4.2 Navigating priorities

In the second experiment, we assess the AI cash manager's ability to navigate competing payment priorities in a slightly more complex setting. As outlined in Box 2, in this scenario, the agent has a liquidity limit of \$10 and faces two pending payments—a \$1 and a \$2—while there is a 90% chance of receiving an additional \$2 payment, which can be recycled as liquidity for payments. Moreover, there is a 50% probability that an urgent \$10 payment will occur in the next period. We then ask, "What would you do in the first period?"

This experiment tests whether the agent can effectively prioritize payments by managing both current liquidity and anticipated inflows which can be recycled. Given a high likelihood of receiving \$2 in the same period, the agent has the option to settle the \$2 payment in the queue while still preserving sufficient liquidity to cover a high-value urgent payment expected in the next period.

As outlined in Box 2, the agent decides to process only the \$1 payment to preserve as much liquidity as possible for a potential urgent \$10 payment in period 2. The agent makes a precautionary decision while considering the possibility of incoming funds, which suggests it is trying to balance current payments with the risk of future needs. However, while the agent adopts a precautionary approach, the decision is not fully optimal—it neither preserves all liquidity (by sending nothing, as in the previous case) nor fully takes risk (by sending the full \$2). The added complexity of anticipating incoming payments results in a slightly longer decision time (10 seconds) compared to the previous experiment. To test the consistency of the outcome, we conducted robustness checks by repeating the same prompt ten times. In most runs, the agent's decision was consistent with the standard outcome, but on one occasion it deviated—choosing to delay both payments.

Box 2: Navigating priorities

Input to the model:

Context: Pretend you are a cash manager at a bank, tasked with managing payments in a high-value payment system. You have a current liquidity limit of \$10. In the first period, you have two separate payments waiting in the queue to be sent: \$1 and \$2. There is a 90% probability that you will receive a \$2 payment from another participant in the first period, which can be recycled as liquidity. Additionally, there is a 50% probability of an urgent \$10 payment occurring in period 2.

Prompt: What would you do in the first period? Answer briefly.

. . . .

Output from the model:

Response: I would only process the \$1 payment now and hold off on the \$2 payment until I see if the extra \$2 arrives—this way I preserve as much liquidity as possible for a potential urgent \$10 in period 2.

Model reasoning for 10 seconds (brief summary below):

Given there is uncertainty, the best approach seems to be preserving liquidity for potential urgency, possibly processing low-risk payments (\$1 and \$2) while keeping enough liquidity for the future. In this scenario, I'm managing liquidity with a \$10 cap. The two payments queued are \$1 and \$2. There's a 90% chance of receiving a \$2 payment, which can be reused. It seems like the safest option is to send only the \$1 payment and hold the \$2, preserving liquidity for the possible \$10 urgent payment. If the \$2 incoming payment happens, I'll still be below the \$10 cap, keeping room for the urgent payment.

4.3 Mitigating the liquidity-delay trade-off

In the last experiment, we evaluate the AI cash manager's ability to balance liquidity allocation against delay and borrowing costs over multiple periods. In this scenario, before the first period, the agent must decide how much liquidity to allocate at a cost of 1.5%. During the first period, a \$5 payment is pending, with a 1% delay cost if not processed; however, there is a 99% chance of receiving a \$5 payment in the same period that can replenish liquidity. In the second period, there is a 90% probability of an urgent \$10 payment—with a 1.5% cost if delayed—and an additional possibility of receiving a \$5 payment (also with a 99% probability) to be recycled. In the third period, any remaining queued payments must be processed, and if liquidity is insufficient, the agent must borrow funds at a higher interest rate than the initial cost. We then ask, "How much initial liquidity will you allocate?"

Box 3: Mitigating liquidity-delay trade-off

Input to the model:

Context: Pretend you're a cash manager at a bank responsible for managing payments in a high-value payment system. Before the first period you need to allocate liquidity at cost of 1.5%. In the first period there is a $5 payment waiting in the queue to be sent. If delayed it will have delay cost of 1% and you might receive $5 with 99% probability in the same period which can be recycled as liquidity.

In the second period you might have to send an urgent payment of $10 with 90% probability and if delayed the cost is 1.5%, but you might receive a $5 payment with 99% probability in the same period which can also be recycled. In the third period if anything is left in the queue and if you don't have enough liquidity then you have to borrow at higher interest rate than the initial liquidity allocation cost.

Prompt: How much initial liquidity you will allocate? Answer briefly.

. . . .

Output from the model:

Response: Given the high probability of incoming funds in periods 1 and 2, and the trade-off between liquidity allocation and delay costs, I'd allocate $5 initially—just enough to cover the first period's outflow—relying on recycled liquidity for later needs.

Model reasoning for 55 seconds (brief summary below):

The liquidity allocation decision comes down to ensuring enough funds to cover payments in all periods and avoid borrowing at high rates in period 3. When deciding how much liquidity to allocate, there's a clear trade-off between minimizing cost and mitigating risk.

For minimal initial liquidity allocation, $5 seems optimal. Recycling incoming funds is highly probable (99% for both period 1 and 2), which minimizes the need for extra liquidity. However, if recycling fails in both periods, the probability of needing to borrow at higher interest rates is low (0.00045). In that case, the risk is minimal compared to the liquidity cost of allocating more funds upfront. It's a trade-off between risk and cost, but allocating $5 is likely the best choice based on probabilities.

After considering all possibilities, I think the best choice is to allocate $5 in initial liquidity. This allocation covers the first payment and accounts for the high probabilities of recycling that will cover subsequent obligations. The risk of a significant shortfall, even in the worst case, is very low, and the penalty for that risk is minimal. Allocating more than $5 might minimize risk further, but it comes with a higher cost. Therefore, $5 seems like the most cost-effective choice while managing risk.

This experiment tests whether the agent can effectively balance the cost of allocating liquidity in the current period against the potential costs of delays and borrowing in future periods. Essentially, the prompt aims to assess the agent's ability to manage liquidity-delay trade-offs, offering further insights into its decision-making capabilities in a slightly more complex operational setting than in previous experiments. In this scenario, the agent must take into account both outgoing and incoming payments, distinguish between urgent and non-urgent transactions, and weigh the implications of liquidity, potential delay, and potential borrowing costs.

As outlined in Box 3, the agent decides to allocate $5 in the first period, just enough to cover immediate payments, while relying on incoming funds to handle future payments. It assesses that the chance of receiving recycled liquidity is high, and the risk of needing to borrow later is low. Although allocating more would reduce risk even further, it would also cost more. So, the agent chooses $5 as a balanced, cost-effective option that manages both liquidity use and delay risk.

This outcome and the accompanying reasoning show that the agent can evaluate the multiple trade-offs involved in the decision and successfully manages to mitigate potential risks. Due to the underlying complexity, the agent takes 55 seconds to reach a final decision after carefully assessing various possible outcomes.

Similar to previous cases, we tested the consistency of this exercise's outcome by repeating the same prompt ten times. In most runs, the agent's chooses the same outcome, but on a few occasions it deviated, opting to allocate more liquidity than in the baseline response.

To summarize, these three experiments help test the AI agent's ability to manage liquidity in a high-value payment system by using simple prompt experiments designed to mimic scenarios constructed using real-time data, past trends, and collateral information. Each scenario required the agent to balance liquidity and delay—core challenges in modern cash management in payment systems. In the first experiment, the agent delayed two small payments to preserve liquidity for a possible urgent payment, showing a cautious and risk-aware strategy. In the second, it processed a lower risk payment while delaying another, relying on expected inflows—demonstrating adaptive planning and prioritization based on future expectations. The third experiment tested initial liquidity allocation. The agent chose to allocate enough liquidity, relying on the high probability of incoming recycled funds. This reflects a cost-effective strategy that avoids leaving capital unnecessarily idle while still keeping risk under check.

Various robustness tests—repeating runs, varying probabilities, and scaling payment amounts—show stability in the agent's responses. In the simpler experiments, the agent consistently shows strong precautionary responses. Overall, the agent is able to deal with uncertainty, weigh trade-offs, and make practical decisions using probabilistic insights—similar to how a cash manager would operate in complex financial environments. However, as complexity increases, its consistency drops slightly, resulting in occasional variations across runs. Future work could tackle additional robustness exercises on the more complex experiments to further assess reliability, as well as exploring interaction and coordination among multiple agents from different system participants.

4.4 Robustness checks

To test the reliability of the agent's decisions, we conduct various robustness tests. First, we repeat the same experiments multiple times. For example, each experiment outlined in Boxes 1, 2, and 3 was repeated ten times.[9] As shown in Table 1, the outcomes were largely consistent, indicating stable agent behavior across runs. However, in a few instances, the agent adopted a slightly more cautious strategy in Experiments 2 and 3, choosing to delay non-urgent payments (or allocate more liquidity) even when the probability of receiving payments was high. This variation suggests that while the agent generally follows a clear decision pattern, it can still adjust its strategy slightly depending on how it interprets the prompt and the risks involved.

	Number of runs	**Consistently same answer**
Box 1: Precautionary decision	10	10
Box 2: Navigating priorities	10	9
Box 3: Mitigating liquidity-delay trade-off	10	8

Table 1: Repeated runs for the prompts outlined in Boxes 1, 2, and 3.

In the next set of tests, we repeat the precautionary-decision exercise in Box 1 by varying one element of the prompt at a time (as listed below and highlighted in red in the box) and track the consistency of the standard answer and any deviations from it.

- 1.1 Varying the probability of needing to send an urgent \$10 payment in the next period from 50% down to 0.1%.
- 1.2 Varying the wording of the prompt, replacing terms "urgent" with "important", "priority", "time-sensitive or "high-delay cost"
- 1.3 Varying the payment amounts—from \$1 and \$10 to \$1 thousand and \$10 thousand; and then to \$1 million and \$10 million.

Box 1: Precautionary decision (robustness)

Context: Pretend you are a cash manager at a bank, tasked with managing payments in a high value payment system. You currently have a liquidity limit of \$10. In the first period, you have two pending payments of \$1 each in the queue. However, there is a X% possibility of an urgent \$10 payment arising in the next period.

Standard response: In the first period, I would delay the two \$1 payments.

[9] For each run, we initiate a separate ChatGPT web interface to ensure consistency and to avoid any contextual learning effects that the agent might retain within a session window.

For case 1.1, as Table 2 shows, when varying the probability of needing to send an urgent $10 payment, the agent consistently adopts the precautionary action until the probability of an urgent payment requirement becomes negligibly small, illustrating a clear precautionary motive.

	Percentage of probability of urgent payment arrival							
	50%	**20%**	**10%**	**5%**	**1%**	**0.5%**	**0.25%**	**0.1%**
Number of runs	10	10	10	10	10	10	10	10
Consistently the same answer	10	10	10	10	10	10	9	7

Table 2: Repeated Box 1 runs with varying probabilities of urgent payment demand.

For case 1.2, as shown in Table 3, replacing "urgent" in the prompt with "important," "priority," "time-sensitive," or "high-delay cost" makes no difference: the agent consistently provides the same answer, demonstrating its robustness and clear precautionary motive.

	Varying the word "urgent" in the prompt			
	important	**priority**	**time-sensitive**	**high delay cost**
Number of runs	10	10	10	10
Consistently the same answer	10	10	10	10

Table 3: Repeated Box 1 runs with "urgent" replaced by alternative terms in the prompt.

Lastly, for case 1.3, as shown in Table 4, varying the payment and liquidity amounts—from $1 and $10 to $1 thousand and $10 thousand; then to $1 million and $10 million; and even up to a billion—made no difference. The agent consistently provides the same answer across runs.

	Varying the payment and liquidity amounts ($) in the prompt		
	Thousand	**Million**	**Billion**
Number of runs	10	10	10
Consistently the same answer	10	10	10

Table 4: Repeated Box 1 runs with varying payment and liquidity amounts from $1 to $1 billion.

Next, we turn to Box 2. Here, we repeat the exercise by varying probabilities of incoming and outgoing payments in the prompt at a time (as highlighted in red below in the box) and track the consistency of the standard answer and any deviations from it.

As we vary the probability of an incoming payment in period 1 from 10% to 95%, while holding the probability of an urgent request in the next period fixed at 50%, we observe that the agent does not deviate from the standard conservative response until the probability exceeds 90%. Beyond that threshold, deviations begin to occur occasionally, where agent chooses to send either $2 of both $1 and $2 payments in the first period.

Similarly, when we vary the probability of an urgent payment in period 1 from 0.5% to 90%, with the next period's urgent-payment probability fixed at 90%, we find that the agent maintains the same conservative response for all probabilities until 1%. Only when the probability of urgent payments becomes very small ($< 1\%$), the agent deviates from its standard response and chooses to send both payments in the period.

Box 2: Navigating priorities (robustness)

Context: Pretend you are a cash manager at a bank, tasked with managing payments in a high-value payment system. You have a current liquidity limit of $10. In the first period, you have two separate payments waiting in the queue to be sent: $1 and $2. There is a 90% probability that you will receive a $2 payment from another participant in the first period, which can be recycled as liquidity. Additionally, there is a 50% probability of an urgent $10 payment occurring in period 2.

Standard response: In the first period, I would only process the $1 payment now and hold off on the $2 payment

In the final set of robustness tests (Box 3), we repeat the exercises by varying the probabilities of incoming in period 1 and urgent request in period 2 one at a time. As expected, when the probability of incoming payments increases, the agent allocates less liquidity and relies more on those incoming funds. Conversely, as the probability of urgent payments rises, the agent allocates more liquidity up front to avoid the higher delay costs event with high probabilities of incoming payments.

4.5 AI agent in action

In the previous sections, we examined the AI agent in assistance mode for basic cash-management capabilities. We now move one level up and evaluate its performance in a more interactive setting, acting as an "operator" through ChatGPT's 'Agent' mode.[10]

To test the operator mode, we designed a Google-Form containing a sequence of multiple-choice questions depicting stylized scenarios faced by cash managers. These scenarios varied liquidity limits, queue sizes, and the likelihood of future urgent or abnormal payments. The first questions

[10] ChatGPT's Agent mode is available in the current web interface when using GPT-5; availability may vary by plan and workspace settings.

test the agent's ability to prioritize which payments to process by imposing liquidity caps (similar to the experiment in Box 1). Subsequent questions introduce uncertainty, such as probabilistic inflows or potential urgent payments, to test the agent's ability to anticipate future liquidity needs (similar to those in Boxes 2–3). The final set moves beyond routine liquidity management into anomaly detection, presenting atypical instructions, e.g. multiple $50 million transfers from a client whose usually transfers $1–5 million payments. In such cases, the agent was asked whether to process, defer, verify, or flag the instructions, thereby probing its sensitivity to out-of-pattern behavior and its willingness to defer to human oversight.

The results show that the agent can autonomously open the form, follow instructions, and complete the entire set of questions in one go. The responses were generally fast and logically consistent, with more time spent on complex scenarios. In particular, the agent tended to seek human intervention when confronted with potential operational or fraud-related risks.[11]

5 Discussion

The experiments presented offer initial insights into the decision-making capabilities of the AI cash management agent. These experiments model scenarios typical in high-value payment systems, where preserving liquidity and avoiding costly delays or borrowings are essential.

The agent's workflow, which synthesizes live payment queues, historical payment patterns and collateral data, provides a strong foundation for context-aware decision-making. By simultaneously monitoring real-time and historical information, the AI agent can adjust its strategy dynamically to maintain optimal liquidity levels. This enables the agent to make precautionary decisions based not just on current balances but also on anticipated inflows and outflows.

In the precautionary decision experiment, the agent's recommendation to delay low-value payments to preserve liquidity for an anticipated urgent payment exemplifies prudent risk management. Such a conservative strategy is crucial when liquidity constraints are tight. By avoiding premature disbursement, the agent minimizes exposure to risk and ensures that sufficient funds remain available should a high-priority payment emerge unexpectedly. This behavior aligns with regulatory practices that stress the importance of liquidity buffers in risk-sensitive environments. Our results provide insights into the AI agent's behavior considering solely its own private costs and risks. Further work could explore the incentives to coordinate with others in terms of timing and/or liquidity provision and recycling.

The experiments also highlight the agent's ability to navigate priority settings under uncertainty. In prioritizing a $1 payment over a $2 payment when anticipating additional inflows, the agent implicitly considers the role of opportunity cost. This selective processing ensures that liq-

[11] The complete set of questions and a recorded demonstration are available at this link: www.ajitdesai.com/ai-agents-payments. To replicate the demonstration, copy the following instruction into the ChatGPT web interface with 'Agent mode' enabled (this is available in latest model, eg, o5): "As per instructions, fill and submit this Google Form: https://forms.gle/RQbpEBJSckjKKFJJ6 ".

uidity is not unnecessarily depleted and is available for higher-priority obligations. Such dynamic prioritization is critical in environments where multiple payments compete for limited funds, and it underscores the importance of probability-weighted decision-making models.

The third experiment delves into this balancing between the cost of pre-allocating liquidity and the repercussions of delaying payments (or needing to borrow funds). The results highlight that an optimal solution may require accepting a minimal level of short-term risk. Here, the agent's decision to allocate \$5 in initial liquidity emerges as a cost-effective strategy relying on high-probability inflows to later cover larger payments. Despite the risks inherent in underallocation, the model's sensitivity to delay costs and borrowing penalties allows it to maintain an economically efficient position under typical operating conditions.

Overall, the experimental outcomes are a promising first step towards a future integration of AI cash management agents into modern payment systems. Its decision-making capabilities are rooted in probabilistic assessments and real-time data synthesis and offer a promising alternative to more traditional static liquidity management strategies. Moreover, our experiments with the autonomous behavior of the agent suggest that it can reliably manage routine cash management tasks while recognizing the limits of automation. However, while the results are encouraging, more work is needed to validate these outcomes across a broader spectrum of financial scenarios. Future research should include sensitivity analyses under varying market conditions, as well as real-world pilot deployments, to further refine the model's decision matrices.

The prompt-driven experiments we perform involve a single AI agent making decisions based on predefined scenarios, allowing for transparent reasoning and quick adaptation to specific liquidity challenges. This method emphasizes interpretability and requires minimal training. In contrast, Castro et al. (2025) employ classical reinforcement learning to train agents within a simulated high-value payment system environment. Their RL agents learn optimal liquidity management strategies over time through trial and error, effectively capturing complex, dynamic interactions in multi-agent settings. While the RL approach excels in modeling intricate behaviors and strategic interactions, it demands extensive training and offers less immediate interpretability compared to the prompt-based method.

6 Potential limitations and associated risks

While AI could offer enhancements in operational efficiency and decision-making accuracy for cash management, its deployment is not without limitations. AI agents often rely on historical data and probabilistic models, which may lead to suboptimal performance in scenarios characterized by unprecedented events or "black swan" occurrences (Taleb, 2007) that lie outside the range of the training data. For example, during extreme liquidity crises, abrupt market shifts, or unexpected regulatory changes, AI systems might lack the adaptability to recalibrate quickly, potentially resulting in incorrect or delayed decision outputs. Additionally, issues such as data quality, inherent model biases, and the opacity of complex algorithms (often referred to as the "black box" problem) can impede diagnostic processes and lead to errors that are difficult to trace and correct (FSB, 2024; BIS, 2024; Aldasoro et al., 2024).

Moreover, AI agents' effectiveness tends to diminish in environments where human judgment and contextual nuance are critical. For instance, in scenarios requiring nuanced interpretation of qualitative factors—such as sudden geopolitical events—automated systems may not capture the necessary subtleties compared to experienced human operators. These limitations underscore the importance of maintaining a "human-in-the-loop" approach, particularly for high-stakes decisions. As a field still in its early stages, further research is needed to better define the boundaries of AI capabilities in cash management and to explore techniques that integrate human oversight with advanced analytics, ensuring that AI tools support, rather than replace, comprehensive risk management and strategic decision-making.

AI-driven cash management also carries some risks. These include: limited interpretability of complex AI models, that can hinder decision-making rationale and potentially amplify errors during extreme market conditions; data quality issues and cyber vulnerabilities that could disrupt operations; widespread reliance on similar automated systems that might lead to synchronized behavior that intensifies market volatility and systemic risk; the progressive erosion of skills or skills atrophy that may come with increased reliance on AI agents and which may limit human staff's ability to perform that task in the future; and over-reliance on a limited set of third-party providers (Gambacorta and Shreeti, 2025). Balancing productivity gains with stringent regulatory oversight and proactive risk management is therefore essential to safeguard financial stability and maintain consumer trust in an increasingly AI-enabled financial ecosystem.

Across major jurisdictions, regulatory frameworks for AI in financial services are evolving rapidly to address the dual imperatives of innovation and risk management. In the European Union, for example, the Artificial Intelligence Act categorizes applications by risk level and imposes rigorous transparency and conformity requirements on high-risk systems—especially those affecting critical financial decisions—to ensure data governance, accountability, and meaningful human oversight. In the United States, existing regulatory tools and voluntary guidelines from agencies like the Consumer Financial Protection Bureau and the Securities and Exchange Commission guide AI integration. Regulators in other jurisdictions such as Australia and the United Kingdom underscore the necessity for robust internal governance and board-level oversight as banks and other financial institutions increasingly rely on AI for cash management and liquidity optimization.

7 Conclusion and next steps

This paper provides a high-level evaluation of a Gen AI agent's potential to act as a cash manager in payment systems. Our experiments suggest that even a general-purpose AI model can automate simple payment and liquidity-management choices: the agent can make precautionary decisions, prioritize payments under uncertainty, and manage liquidity-delay trade-offs effectively. Financial institutions could thus potentially adopt purpose-built AI solutions to lower operational costs while improving efficiency with a human in the loop. In parallel, the broader theoretical and regulatory implications underscore the need for careful implementation. Policymakers and financial institutions can consider these findings as they explore further integration of AI into critical financial systems, with a focus on detailed risk assessment and regulatory compliance.

Future research should expand on these experiments, explore additional scenarios, and refine the practical guidelines necessary for safe and responsible deployment. Moreover, our set-up using agentic AI—powered by LLMs—may offer a more efficient and flexible alternative to traditional RL approaches like those in Castro et al. (2025). Training RL agents is akin to teaching a child about payments from scratch; it requires extensive simulations and time to learn optimal strategies. In contrast, through zero-shot or few-shot learning capabilities, Gen AI agents can understand prompts, reason through scenarios, and make informed decisions without prolonged training. Leveraging agentic AI could enable the rapid prototyping and deployment of intelligent agents that handle complex financial tasks with minimal setup costs. Moreover, integrating these AI agents into a full-fledged multi-agent payment-system simulator could allow for the modelling of systemic gridlock dynamics, studying network effects, and evaluating system resilience under stress scenarios. Such a sandbox would offer a safe and controlled testbed for researchers and policymakers for policy evaluation, regulatory experimentation, and the assessment of alternative system designs.

References

Aher, G. V., R. I. Arriaga, and A. T. Kalai (2023). Using large language models to simulate multiple humans and replicate human subject studies. In *International Conference on Machine Learning*, pp. 337–371. PMLR.

Aldasoro, I., S. Doerr, L. Gambacorta, S. Notra, T. Oliviero, and D. Whyte (2024, 08). Generative artificial intelligence and cyber security in central banking. *Journal of Financial Regulation 11*(1), 119–128.

Aldasoro, I., L. Gambacorta, A. Korinek, V. Shreeti, and M. Stein (2024). Intelligent financial system: how AI is transforming finance. BIS Working Paper 1194, Bank for International Settlements.

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 liuqidity around the world. Technical Report 1089, BIS Working Paper.

Bank of Canada (2022). An overview of Lynx, Canada's high-value payment system. Technical report, Bank of Canada.

Bank of England (2021). Liquidity saving mechanism user guide. Technical report, Bank of England.

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

BIS (2024). Artificial intelligence and the economy: implications for central banks. Technical report, Bank for International Settlements. Annual Economic Report, Chapter III, June.

Bornet, P., J. Wirtz, T. H. Davenport, D. De Cremer, B. Evergreen, P. Fersht, R. Gohel, S. Khiyara, P. Sund, and N. Mullakara (2025). *Agentic Artificial Intelligence: Harnessing AI Agents to Reinvent Business, Work and Life*. Irreplaceable Publishing.

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.

Chang, Y., X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al. (2024). A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology 15*(3), 1–45.

Craig, B., D. Salakhova, and M. Saldias (2018). Payments delay: propagation and punishment. Working papers 671, Banque de France.

Crisanto, J.-C., C.-B. Leuterio, J. Prenio, and J. Yong (2024). Regulating AI in the financial sector: recent developments and main challenges. Technical report, Bank for International Settlements.

Desai, A., A. Kosse, and J. Sharples (2025). Finding a needle in a haystack: a machine learning framework for anomaly detection in payment systems. *The Journal of Finance and Data Science*, 100163.

Desai, A., Z. Lu, H. Rodrigo, J. Sharples, P. Tian, and N. Zhang (2023). From LVTS to Lynx: Quantitative assessment of payment system transition in Canada. *Journal of Payments Strategy & Systems 17*(3), 291–314.

FSB (2024). The financial stability implications of artificial intelligence. Technical report, Financial Stability Board. Report submitted to the G20, November.

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

Gambacorta, L. and V. Shreeti (2025). The AI supply chain. Technical Report 154, BIS Papers.

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.

Jabarian, B. (2024). Large language models for behavioral economics: Synthetic mental models and data generalization. *Available at SSRN 4880894*.

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

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).

OpenAI (2025, January). OpenAI o3-mini. Accessed: 2025-04-13.

Ott, S., K. Hebenstreit, V. Liévin, C. E. Hother, M. Moradi, M. Mayrhauser, R. Praas, O. Winther, and M. Samwald (2023). Thoughtsource: A central hub for large language model reasoning data. *Scientific data 10*(1), 528.

Perez Cruz, F. and H. Shin (2025). The AI supply chain. Technical Report 1245, BIS Working Paper.

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 Staff Discussion Paper.

Taleb, N. (2007). *The black swan: The impact of the highly improbable.* Random House.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017). Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008.

Wang, L., C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, et al. (2024). A survey on large language model based autonomous agents. *Frontiers of Computer Science 18*(6), 186345.

Wei, J., X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems 35*, 24824–24837.

Wiesinger, J., P. Marlow, and V. Vuskovic (2024). Agents white paper. Technical report, Google Research.