informs
https://pubsonline.informs.org/journal/mnsc

MANAGEMENT SCIENCE
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0025-1909, EISSN 1526-5501

Submitted to *Management Science*

Optimizing Service Operations via LLM-Powered Multi-Agent Simulation

(Authors' names are not included for peer review)

Abstract. Service system performance depends on how participants respond to design choices, but modeling these responses is hard due to the complexity of human behavior. We introduce an LLM-powered multi-agent simulation (LLM-MAS) framework for optimizing service operations. We cast the problem as stochastic optimization with decision-dependent uncertainty: design choices are embedded in prompts and shape the distribution of outcomes from interacting LLM-powered agents. By embedding key numerical information in prompts and extracting it from LLM-generated text, we model this uncertainty as a controlled Markov chain. We develop an on-trajectory learning algorithm that, on a single simulation run, simultaneously constructs zeroth-order gradient estimates and updates design parameters to optimize steady-state performance. We also incorporate variance reduction techniques. In a sustainable supply chain application, our method outperforms several benchmarks, including black-box optimization and using LLMs as numerical solvers or as role-playing system designers. A case study on optimal contest design with real behavioral data shows that LLM-MAS is both a cost-effective evaluator of known designs and an exploratory tool that can uncover viable designs overlooked by traditional approaches.

Key words: LLM, multi-agent simulation, controlled Markov chain, zeroth-order optimization, service operations

1. Introduction

Service systems are driven by decisions and interactions of people. Examples include multi-echelon supply chains (Cohen et al. 2016), innovation contests (Chen et al. 2020), and digital marketplaces (Bandi et al. 2024). A central task in these settings is to design policies, incentives, and operational levers that deliver strong performance. These design parameters, including sustainability rules, prize structures, and platform interfaces, shape how individuals decide and communicate. Individual choices, such as adopting a regulation, entering a contest, or selecting a product, then aggregate into outcomes such as cost effectiveness, efficiency, and externalities. Yet human behavior often deviates from analytical and empirical predictions, because decisions reflect how people form beliefs, assess risks, and make trade-offs in contexts that are hard to observe. Effective designs should therefore account for behavioral factors to capture how people respond to design choices, as these responses ultimately determine system performance (Huang et al. 2013).

Despite broad recognition of their importance in operations management (OM), human decision processes remain hard to model (Donohue et al. 2020). Analytical models, such as utility-based formulations, use simplified assumptions and a small set of parameters to maintain tractability. This helps abstraction but often misses key features of behavior. Calibrating these models for practical use also requires data that are costly to collect in OM settings, whether through lab studies or field experiments. A further challenge is that behavior

is not fully rational. Decisions are shaped by cognitive limits and by time and attention constraints during the decision process (Donohue et al. 2018). These factors reduce the accuracy of traditional modeling and analysis approaches and limit their usefulness for system design in practice.

We take a different approach. Instead of building analytical models and calibrating them with costly data, we use large language models (LLMs) to simulate human decisions. Pretrained on extensive human-generated text, LLMs capture statistical patterns in how people reason, communicate, and make decisions. Recent studies report human-like behavior from LLMs across economics (Chen et al. 2023), marketing (Goli and Singh 2024), and political science (Argyle et al. 2023). Most relevant to our paper, Chen et al. (2025) show that GPT models largely reproduce the decision biases as documented by Davis (2018) in canonical OM settings (e.g., inventory management, sourcing, forecasting, and queueing).

Beyond generating human-like behavior (the response), LLMs can read unstructured, narrative-rich information that traditional numerical simulators struggle to handle (the comprehension). Many operational outcomes hinge on descriptive or strategic text such as advertisements, customer reviews, recommender explanations, and chatbot replies. As native text processors, LLMs can both analyze and generate language that responds to nuances in phrasing, sentiment, and dialogue context. This opens new possibilities for experimentation on system designs in digital operations such as review-informed pricing, customer-service automation, and AI-mediated marketplaces.

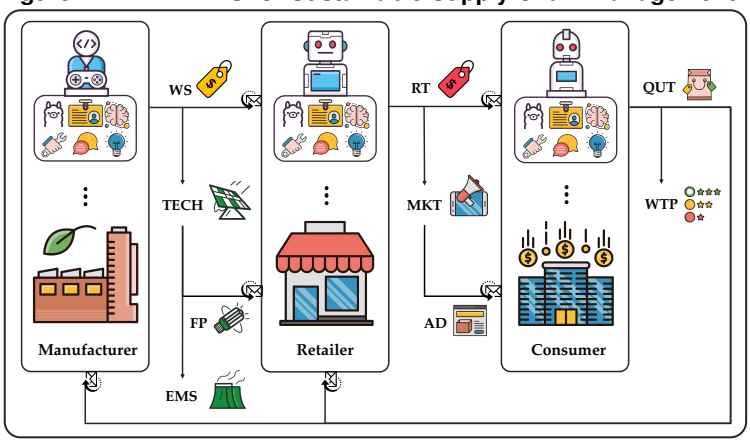
We treat LLMs as agents within a service system and let them interact under the given system design. Their interactions generate system-level outcomes, which determine the design's performance. We refer to this approach as LLM-powered multi-agent simulation (LLM-MAS). It follows the agent-based modeling paradigm, where micro-level behaviors aggregate into macro-level dynamics, making it well-suited for systems with complex interdependencies (Axtell and Farmer 2025).[1]

We illustrate the LLM-MAS approach to system design with the following example. As AI reshapes supply chain management (Cohen and Dai 2026), a policymaker seeks to develop sustainable policies that balance economic viability, environmental impact, and overall supply chain well-being (Krass et al. 2013, Cohen et al. 2016). This is our running example. We also present in Section 7 a case study on the optimal design of innovation contests, drawing on real behavioral data.

EXAMPLE 1 (SUSTAINABLE SUPPLY CHAIN MANAGEMENT). Consider a three-echelon supply chain with manufacturer (M), retailer (R), and consumer (C); see Figure 1. The policymaker designs sustainable policies with two parameters: a carbon tax on the manufacturer (θ_1) and a consumer subsidy (θ_2). For any given policy $\theta = (\theta_1, \theta_2)$, we simulate the decisions of all three players using LLM-MAS rather than

[1] Traditional agent-based models rely on handcrafted rules or simple functions to represent behavior and interaction. These rule-based approaches often oversimplify decision-making and generalize poorly across contexts. Recent advances apply reinforcement learning (RL) to train agents that learn more complex decision rules. However, RL assumes reward maximization, conflicting with the evidence that people are not perfect optimizers (Allon et al. 2023). Moreover, defining reward functions that faithfully capture human decision processes remains challenging. These limitations motivate the use of LLMs as more expressive agent models.

Figure 1 LLM-MAS for Sustainable Supply Chain Management



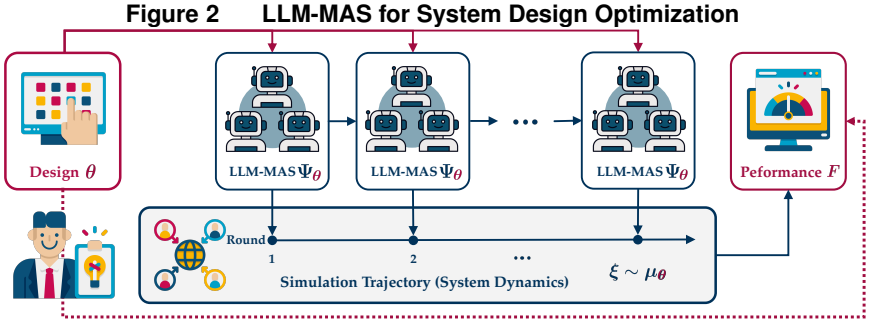
specifying their behavior analytically. The manufacturer chooses a wholesale price (WS) and an investment level in low-carbon technology (TECH); the latter determines the system's carbon emissions (EMS). The manufacturer also discloses the carbon footprint (FP) to the retailer, which is defined as the percentage reduction in carbon emissions relative to a baseline. The retailer sets the retail price (RT) and the marketing budget (MKT), then combines the price with the carbon footprint to create an advertisement (AD) funded by the budget. After viewing the ad, the consumer chooses purchase quantities (QUT) and reveals willingness to pay (WTP). The ad narratives are generated by an auxiliary LLM, and the consumer interprets the ad and acts, highlighting the strength of LLM-MAS at processing unstructured information end to end. The system state (ξ) includes all agents' decisions (i.e., $\texttt{TECH}, \texttt{WS}, \texttt{MKT}, \texttt{RT}, \texttt{QUT}, \texttt{WTP}$) and the resulting outcome (i.e., EMS). The policymaker evaluates each policy using $F(\theta;\xi) = -\big(\texttt{SCWF}(\theta;\xi) - \texttt{FISC}(\theta;\xi) - \texttt{ENV}(\xi)\big)$, which balances supply chain welfare (SCWF), fiscal costs (FISC), and environmental externalities (ENV). The goal is to choose policy parameters that optimize long-run system performance over multiple rounds of interactions among the three agents.

Inspired by this example, we pose the following question:

"How can LLM-MAS be integrated into system design optimization?"

To address this question, we propose a principled framework (Figure 2). A system designer seeks to optimize design choices using the dynamics generated via LLM-MAS, where LLM-powered agents serve as *silicon proxies*. This framework casts the problem as stochastic optimization with uncertainty driven by LLM-generated responses, facilitates the development of optimization algorithms, and accommodates any off-the-shelf LLM. We use these models as is, assuming that LLMs can accurately simulate human behavior in the application of interest, so we do not fine-tune or post-train models within LLM-MAS. (Calibrating models to improve behavioral fidelity is outside the scope of this paper.) This assumption is supported by our case study on optimal contest design, where the LLM-MAS, implemented with an open-source model, effectively reproduces the lab experiments of Huang et al. (2026) involving hundreds of human participants.

Figure 2 LLM-MAS for System Design Optimization



Note. (i) Ψ_θ is the one-round simulation of system state ξ. (ii) μ_θ is the stationary distribution for a given design θ.

This framework serves as a digital twin that links theory to practice and complements existing approaches to system design optimization along two dimensions. (i) *Evaluation (bridging theory and reality).* It narrows the gap between elegant theoretical models and complex real systems. By testing theoretical solutions in LLM-MAS, it relaxes idealized assumptions and checks robustness under realistic conditions. This is particularly valuable in domains where human behavior shapes outcomes, clean models fall short, and behavioral data are costly to collect. (ii) *Exploration (providing a silicon sandbox).* It enables structured exploration of alternative designs in a controlled, synthetic environment. The testbed helps find designs that perform well when human behavior plays a crucial role, which is often analytically intractable, while reducing the time and resources needed for lab experiments.

1.1. Main Contributions

1.1.1. Modeling. We formulate system design via LLM-MAS as a stochastic optimization problem with decision-dependent uncertainty, represented as a controlled Markov chain. Design parameters enter as prompt variables for each LLM-powered agent, which shift the distribution of token outputs. This token-level randomness yields text responses that encode each agent's behavior, and over multiple rounds of interactions, these behaviors aggregate to determine system performance. We construct a system-level Markovian state by identifying compact, meaningful variables from sequences of LLM prompts and responses that constitute the agents' observations, communications, and actions. This representation clarifies how design choices drive the evolution of uncertainty in LLM-MAS and provides a foundation for efficient learning algorithms for system design optimization.

1.1.2. Algorithms. We develop a new algorithm to address two challenges from the resulting stochastic optimization problem. The first is gradient computation. With decision-dependent uncertainty, the gradient depends on the conditional distribution of the system state given the design. In LLM-MAS, this distribution is intractable. The uncertainty varies with design parameters in complex ways and arises from numerous sources. We use a zeroth-order method that constructs gradient estimates from function evaluations, which removes the need to explicitly model the conditional distribution.

The second challenge concerns steady-state evaluation. A direct approach requires many trajectories of the controlled Markov chain, each with many rounds of agents' interaction, to approximate the steady-state distribution. This is computationally prohibitive considering LLM inference latency. We propose an on-trajectory learning algorithm which, within a single simulation run, simultaneously estimates gradients and updates design parameters, but on two timescales for convergence. It learns from the full sample path, instead of just the terminal state as in conventional multi-trajectory methods, thereby reducing LLM queries.

On the theory side, we establish the convergence of our proposed algorithm through the lens of stochastic approximation (SA) with adaptive Markovian data. This setting arises because the transition distribution is not fixed but depends on the current iterate, which itself is updated based on the data collected thus far. We adopt a Poisson equation approach to address the temporal dependence present in the data. By leveraging the smoothness of its solution, we bound the gap between the zeroth-order estimator and stochastic gradient under the transition distribution while controlling the deviation from the steady state.

Furthermore, we incorporate two variance reduction techniques to enhance the algorithm's efficiency: one exploits the known structure of the performance function (guided perturbation), while the other utilizes information from the algorithm's iterates when such structure is unavailable (residual feedback).

1.1.3. Empirics. We demonstrate the practical value of the LLM-MAS framework through two applications: sustainable supply chain management and innovation contest design. In the supply chain setting, our methods are benchmarked against Bayesian optimization, LLM-as-a-Solver, and LLM-as-a-Designer. Across all benchmarks, our algorithm and its variance-reduction variant achieve superior performance. The results offer two key insights. First, modeling uncertainty as a controlled Markov chain exploits the probabilistic structure of LLM-MAS for greater computational efficiency. Second, relying on LLMs alone is insufficient for system design; effective solutions require rigorous stochastic formulations and problem-specific, numerical optimization algorithms. These findings align with the growing view in supply chain management that disruptive GenAI tools are complements, not substitutes, for traditional OM methods (Cohen and Dai 2026).

In the contest design setting, we use behavioral data from a recent lab experiment (Huang et al. 2026) to show how LLM-MAS can both evaluate existing designs and search for improved ones. This is especially important when human behavior matters. LLM-MAS closely replicates laboratory heterogeneity in entry and effort decisions across designs with varying entry fees, reserves, and shared prizes. Therefore, it serves as a cost- and time-efficient alternative to conducting new lab experiments for design evaluation. Notably, both lab and LLM-MAS results deviate from game-theoretic predictions assuming fully rational, risk-neutral agents, further suggesting the importance of incorporating behavioral realism into system design.

A key by-product of LLM-MAS is the rich reasoning-process data that reveal the agents' decision logic, which offers a level of interpretability rarely attainable with other AI tools such as RL or with lab/field experiments lacking post-hoc interviews. For instance, reasoning traces from our contest-design case study

can help designers assess whether a design continues to elicit effort as the liability cap scales. This, in turn, supports more reliable evaluation of incentive mechanisms.

Furthermore, we optimize contest design. The LLM-MAS solution differs markedly from the game-theoretic recommendation, which ties the reserve and shared prize to the entry fee and favors higher entry fees. In contrast, our solution decouples the three design parameters, recommends a moderate entry fee, and sets both the reserve and shared prize well below the game-theoretic counterparts. It yields a much higher total effort (the objective) than the theoretical optimum. These results indicate that LLM-MAS is capable of identifying viable designs overlooked by analytical models and of informing future empirical studies.

1.2. Related Works

1.2.1. LLM-MAS. Our work connects to a fast-growing literature that uses LLMs to simulate and study human behavior in business, economics, and social sciences (Gao et al. 2024). The appeal is clear: LLMs can lower the cost of collecting behavioral data by orders of magnitude and enable experiments where human studies are otherwise infeasible or ethically constrained. While not perfectly replicating human behavior, LLM-generated data still help test new ideas. A key thread in this literature builds high-fidelity simulation environments to answer what-if questions, and thereby support counterfactual analysis, hypothesis testing, and theory prototyping (Park et al. 2022). For instance, Allouah et al. (2025) build an LLM-powered e-commerce sandbox to examine how operational levers and product attributes influence buyers' actions, providing implications for both platform operators and sellers. Our contribution advances this line of research from *design evaluation* to *design optimization*: we use LLM-MAS across design choices not only to assess outcomes but also to optimize performance.

1.2.2. Intersection of LLMs and OR. This research aligns with recent efforts at the intersection of LLMs and operations (Dai and Swaminathan 2026). Prior work broadly follows two directions: LLMs for operations research (OR) and OR for LLMs. Most studies in the first direction use LLMs for mathematical modeling by translating natural-language problem statements into mathematical formulations, with applications to deterministic optimization (Huang et al. 2025), robust optimization (Bertsimas and Margaritis 2024), and dynamic programming (Zhou et al. 2025). Our study belongs to this stream but differs in purpose. Instead of using LLMs to convert text into stochastic programs for generic numerical solvers, we formulate the mathematical model ourselves, use LLMs to simulate uncertain dynamics, and solve the problem with a tailored algorithm that reflects LLM-specific features.

The second direction, OR for LLMs, applies OR methods to improve LLM performance, treating LLMs as an application domain instead of as tools for OR. This line of research primarily addresses inference-time efficiency, typically through scheduling and batching to manage the dual queues of token prefill and decoding (Jaillet et al. 2025, Li et al. 2025); see also Mitzenmacher and Shahout (2025) for a recent survey. Our framework also accounts for the distinctive characteristics of LLMs, but the goal is to deploy them for simulation-based optimization rather than to enhance their internal capabilities.

1.2.3. Zeroth-Order Optimization and SA. The proposed algorithm falls under zeroth-order methods for stochastic optimization when gradients are unavailable or costly to compute. These methods update solutions much like gradient-based approaches but estimate gradients using function evaluations; see LA and Bhatnagar (2025) for an introduction. Most prior studies focus on settings where the distribution of the random variable does not depend on the decision. In contrast, our optimization problem has decision-dependent uncertainty (Drusvyatskiy and Xiao 2023), captured by the stationary distribution of a controlled Markov chain, which introduces complex intertemporal dependencies into the algorithm.

Analyses of zeroth-order methods typically draw on SA (Kushner and Yin 2003). We follow this practice, but our on-trajectory learning algorithm gives rise to SA with adaptive Markovian data. This type of SA has gained much attention due to its application to RL algorithms with policy improvement (Konda and Tsitsiklis 2003, Che et al. 2026, Li et al. 2026). However, these studies usually assume access to first-order information, in contrast to our setting where gradients are intractable and must be estimated using zeroth-order methods.

The remainder of this paper is organized as follows. Section 2 formulates system design via LLM-MAS as a stochastic optimization problem with decision-dependent uncertainty driven by a controlled Markov chain. Section 3 explains how to extract the Markov chain's state variables from text exchanges between LLM-powered agents. Section 4 presents the on-trajectory learning algorithm to address challenges specific to LLM-MAS and establishes its asymptotic convergence. Section 5 incorporates two variance reduction techniques to enhance efficiency. Section 6 presents numerical experiments in a supply chain setting to compare algorithm performance. Section 7 provides a case study on optimal contest design using real human behavioral data. Section 8 concludes, and all technical proofs are deferred to the e-companion.

2. Problem Formulation

Consider a service system with design parameter vector $\theta \in \Theta \subseteq \mathbb{R}^d$. We evaluate its performance through LLM-MAS. Let $\xi \in \Xi$ be the random variable that captures the uncertainty in the system. The system performance is measured by a cost function $F(\theta;\xi)$. Our goal is to find an optimal design that minimizes the expected cost with respect to the distribution of ξ.

In the supply chain example (Example 1), the design θ consists of a carbon tax on the manufacturer and a consumer subsidy. The uncertainty ξ faced by the system designer includes agent behavior and system outcomes: the manufacturer's wholesale price and investment in low-carbon technology, the retailer's retail price and marketing budget, the consumer's purchase quantities and willingness to pay, and the system's carbon emissions. The cost function F reflects supply chain welfare, fiscal spending, and environmental externalities.

What sets this problem apart from most stochastic programs is how the distribution of ξ is generated and how it depends on θ. In LLM-MAS, θ is embedded in the prompts, shapes token probabilities, affects agent

actions, and alters system outcomes. As a result, the distribution of ξ varies with θ. We therefore formulate the system design problem as stochastic optimization with *decision-dependent* uncertainty:

$$\min_{\theta\in\Theta}\left\{f(\theta) \coloneqq \mathbb{E}_{\xi\sim\mu_\theta}[F(\theta;\xi)]\right\}, \tag{1}$$

where μ_θ denotes the distribution of ξ for a given design θ. Here, the term "decision" refers to the system designer's choice θ, rather than the agents' actions simulated by LLMs. In what follows, we detail the mechanism underlying the uncertainty within LLM-MAS.

2.1. Token-Level Uncertainty

LLMs produce random outputs because they are autoregressive probabilistic models. At inference time, the model predicts a probability distribution over the next token given the preceding context. It does so by computing logits over its vocabulary (i.e., the fixed set of output tokens, typically subword units), applying a softmax transformation to derive probabilities, and selecting the token according to a decoding rule.

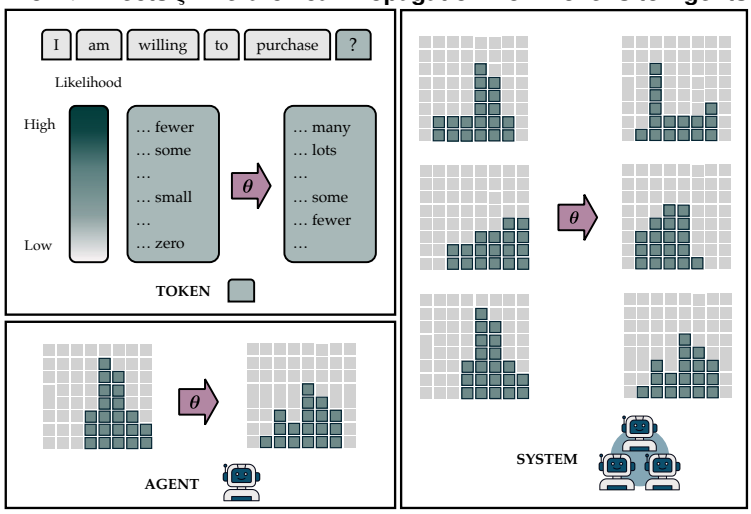
In practice, decoding is rarely deterministic. Modern LLMs such as GPT, LLaMA, and DeepSeek decode tokens probabilistically, with sampling parameters like temperature controlling randomness in the next-token generation process (Raschka 2024). For example, higher temperatures flatten softmax distribution, leading to greater variability in generated tokens. Even when two runs of simulation are initialized with the same prompt, small sampling differences in early tokens can alter subsequent context and propagate through later token distributions. As a result, the two runs gradually drift apart as the token sequence unfolds.

REMARK 1. Even when the temperature is nominally set to zero, identical LLM inferences are not guaranteed due to several non-deterministic factors. (i) When LLMs are quantized to low-precision for efficient deployment, the coarser representation increases the number of near-ties, causing tie-breaking to occur more frequently (Yuan et al. 2025). Such tie-breaks may vary between runs, with token flips sometimes cascading into entirely different completions. (ii) Loading a sharded LLM across GPUs can expose the non-associativity of floating-point arithmetic. Operations such as summation may execute in different orders, introducing rounding discrepancies that can, in turn, flip the argmax when logits are nearly equal (Dao et al. 2022). (iii) Third-party LLM services providing access to proprietary models such as GPT and Claude often batch prompts to boost throughput. Over time, small variations in batch composition can nudge the model toward noticeably different outputs (Rajbhandari et al. 2022).

2.2. Decision Dependence

In LLM-MAS, each agent's behavior is simulated by prompting an LLM with the information that a human participant in an equivalent setting would have. The prompt specifies the current system design θ, the agent's persona, relevant behavior of other agents, and any private context. The LLM then processes this prompt and generates tokens that describe the agent's behavior. Agents subsequently interact following the sequence defined by the simulation environment, which determines the event order and the information flow. Finally,

Figure 3 How θ Affects ξ: Hierarchical Propagation from Tokens to Agents to System



Note. The distributions are illustrative only. A higher consumer subsidy (θ_2) embedded in the prompt shifts token likelihood following "I am willing to purchase ..." from lower- (e.g., "fewer", "some") to higher-intensity ones (e.g., "many", "lots").

the simulated behaviors are aggregated into a system-level outcome ξ, which contains the quantities used in the performance function F.

Because θ is embedded in each prompt, changing θ alters the token probabilities. These alterations, in turn, change the distribution of agent actions extracted from the LLM outputs. Through agent interactions, these changes further accumulate into the system state ξ. This hierarchical propagation explains how incorporating θ in prompts induces a system-level distribution μ_θ over ξ.

To illustrate, consider the supply chain setting in Example 1. Consumer subsidy (θ_2), one of the design parameters, is included in the prompt provided to the consumer agent: "`You receive a subsidy of SUBSIDY dollars for each unit of product you purchase.`" Here, `SUBSIDY` is replaced with the actual value of θ_2 when prompting an LLM to generate the consumer's purchasing behavior. A larger θ_2 makes the agent more likely to produce tokens that reflect higher purchase volume, which consequently changes the distribution of the system-level outcome ξ. See Figure 3 for details.

2.3. Steady-State Performance

When using LLM-MAS to simulate and optimize service systems, the system state ξ typically evolves over multiple rounds of interactions among agents, and should be treated as a stochastic process, taking the value $\xi^{(t)}$ in round t. This dynamic mirrors reality: human participants need time to interpret new rules, observe others' responses, and adjust their own strategies. Only after a period of adaptation do collective behaviors converge to stable patterns, at which point the effects of a system design reveal its long-term performance.

Accordingly, we focus on steady-state performance in this paper. A long-run perspective captures the adaptation dynamics and delineates the distribution of outcomes after the system has equilibrated. Moreover, once a service design is finalized, subsequent modifications are difficult to implement without disrupting

ongoing operations. Optimizing steady-state behavior *ex ante* can reduce the risk of costly reconfigurations and mitigate operational instability. In the next section, we show that, by appropriately extracting information from LLM outputs that encode agents' behavior, $\xi^{(t)}$ can be modeled as a controlled Markov chain. Consistent with this focus, μ_θ in (1) represents the stationary distribution of the chain for a given θ. (The methodology developed in this paper also applies to finite-time performance, including problems with fixed operational horizons or round-based settings such as contests and auctions.)

3. LLM-MAS as Controlled Markov Chain

We begin with a general formulation of LLM-MAS and specify key elements from the agents to define the system state $\xi^{(t)}$: agents' observations of the environment, inter-agent messages, and agents' actions. We then revisit the supply chain example to show how to identify these elements in practice. Next, we detail how to prompt LLMs, with particular emphasis on incorporating numerical quantities in the inputs and specifying the required outputs. This setup allows the system state to be extracted from textual LLM outputs round by round, completing the update from $\xi^{(t)}$ to $\xi^{(t+1)}$.

3.1. System State

An LLM-MAS system is represented by the tuple $\langle \mathcal{G}, \Psi_\theta, T \rangle$: a directed graph $\mathcal{G}$, a one-round simulation operator Ψ_θ, and a time horizon T. The graph $\mathcal{G}$ is defined by an index set $\mathcal{N}$ of agents and a matrix $\mathcal{M}$ that encodes directed interaction links, typically representing information, product, or cash flows.

Each agent $i \in \mathcal{N}$ has five modules: profile, perception, communication, LLM, and memory. In each simulation round $t = 1, \ldots, T$, these modules operate as follows and the order of events is shown in Figure 4.

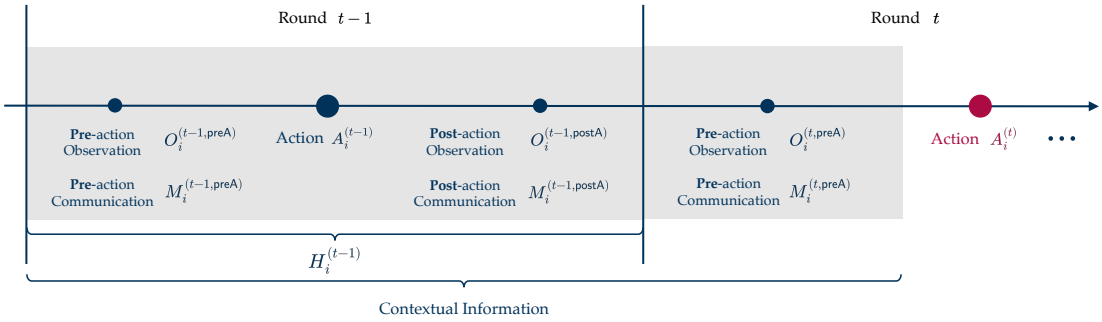
(i) Profile. Defines the agent's functional role $\mathcal{R}_i$ and, optionally, heterogeneous attributes χ_i that capture individual characteristics (e.g., risk aversion) and social preferences (e.g., propensity to collaborate).

(ii) Perception. Specifies the agent's observation $O_i^{(t)}$. It reflects interactions with the environment, not with other agents. The observation may include individual information (available only to the agent) and/or shared information (available to all agents). In the supply chain example, the manufacturer's carbon emissions are part of the shared observation. In an auction, a bidder's valuation of the item being offered is part of the individual observation.

(iii) Communication. Captures the agent's interactions with others. Let $M_i^{(t)} = \bigcup_{j \in \mathcal{M}[i]} (M_{i \to j}^{(t)}, M_{i \leftarrow j}^{(t)})$ denote the collection of information extracted from the exchanges over its links, including outbound messages $M_{i \to j}^{(t)}$ (sent to j) and inbound messages $M_{i \leftarrow j}^{(t)}$ (received from j) for $j \in \mathcal{M}[i]$, the set of agents with which agent i communicates. Each message may contain numerical or textual content.

(iv) LLM. Serves as the agent's behavior generator, producing the action $A_i^{(t)}$ based on the system design θ and the contextual information. In each round, the agent may obtain new information from the environment and other agents, both before and after acting. Only information available before the action is provided to the LLM as context for reasoning. This context also includes information from the

Figure 4 Order of Events



Note. Contextual information is the inputs, beyond the system design θ, that are provided to the LLM to simulate the agent's action.

previous round. Information that arrives after the action is recorded and carried to the next round via the memory module. Accordingly, we split the observation $O_i^{(t)}$ into a pre-action part $O_i^{(t,\text{preA})}$ and a post-action part $O_i^{(t,\text{postA})}$. We apply the same practice for communications: $M_i^{(t)} = (M_i^{(t,\text{preA})}, M_i^{(t,\text{postA})})$.

(v) Memory. Stores this round's new information $H_i^{(t)} = (O_i^{(t)}, M_i^{(t)}, A_i^{(t)})$, including the agent's observation, communication, and action.

The system state at the end of round t is the collection of new information from all agents: $\xi^{(t)} = \bigcup_{i\in\mathcal{N}} H_i^{(t)}$. We emphasize that components of $O_i^{(t)}$, $M_i^{(t)}$, and $A_i^{(t)}$ are typically numerical, possibly containing textual content. They should be properly embedded within the LLM prompts and subsequently extracted from the LLM outputs (see Section 3.3). The state transition to the next round is governed by the one-round simulation operator Ψ_θ, which maps $\xi^{(t)}$ to $\xi^{(t+1)}$ under the system design θ.

Iterating Ψ_θ over the horizon T yields the trajectory $\{\xi^{(t)} : t = 0, 1, \ldots, T\}$, where each state depends only on its immediate predecessor. Thus, the LLM-MAS dynamics are Markovian. With θ fixed along the trajectory, the system state forms a controlled Markov chain, since the distribution of agents' behavior simulated by LLMs is affected by θ (see Section 2.2).

3.2. Illustration with the Supply Chain Example

In Example 1, the system has three agents with different roles: manufacturer (M), retailer (R), and consumer (C). Consider the retailer. In each round t, its action $A_\text{R}^{(t)}$ consists of a retail price $\texttt{RT}^{(t)}$ and a marketing budget $\texttt{MKT}^{(t)}$. These are chosen using the previous round's memory $H_\text{R}^{(t-1)}$ and the pre-action message from the manufacturer $M_{\text{R}\leftarrow\text{M}}^{(t,\text{preA})}$. After acting, the retailer calls a tool (simulated by an auxiliary LLM) that uses $\texttt{RT}^{(t)}$ and the footprint $\texttt{FP}^{(t)}$ to generate the ad narratives $\texttt{AD}^{(t)}$ under the budget $\texttt{MKT}^{(t)}$. It then sends $(\texttt{RT}^{(t)}, \texttt{AD}^{(t)})$ to the consumer. After the consumer acts, the retailer receives the purchase quantity $\texttt{QUT}^{(t)}$.

The retailer has no pre-action observation, so $O_\text{R}^{(t,\text{preA})} = \varnothing$. Its pre-action communication is the message from the manufacturer $M_{\text{R}\leftarrow\text{M}}^{(t,\text{preA})} = (\texttt{WS}^{(t)}, \texttt{FP}^{(t)})$. Its post-action communications include the message sent

Table 1 Components of System State in Example 1

$\mathcal{R}$	$O^{(t,\mathsf{preA})}$	$M^{(t,\mathsf{preA})}$		$A^{(t)}$	$O^{(t,\mathsf{postA})}$	$M^{(t,\mathsf{postA})}$	
		sending	receiving			sending	receiving
M	$\varnothing$	$\varnothing$	$\varnothing$	WS, TECH	EMS	WS, FP (to R)	QUT (from R)
R	$\varnothing$	$\varnothing$	WS, FP (from M)	MKT, RT	$\varnothing$	AD, RT (to C); QUT (to M)	QUT (from C)
C	$\varnothing$	$\varnothing$	AD, RT (from R)	WTP, QUT	$\varnothing$	QUT (to R)	$\varnothing$

Note. WS: wholesale price; TECH: level of investment in low-carbon technology; MKT: marketing budget; RT: retail price; WTP: willingness to pay; QUT: purchase quantities; EMS: carbon emissions; FP: carbon footprint; AD: advertisement narratives. See Example 1 and Figure 1 for details.

to the consumer $M_{\mathsf{R}\to\mathsf{C}}^{(t,\mathsf{postA})} = (\mathtt{RT}^{(t)}, \mathtt{AD}^{(t)})$ and the message received in return $M_{\mathsf{R}\leftarrow\mathsf{C}}^{(t,\mathsf{postA})} = \mathtt{QUT}^{(t)}$ after the consumer acts. There is no post-action observation. Thus, at the end of this round, its memory module stores

$$H_{\mathsf{R}}^{(t)} = (\underbrace{\mathtt{WS}^{(t)}, \mathtt{FP}^{(t)}}_{M_{\mathsf{R}\leftarrow\mathsf{M}}^{(t,\mathsf{preA})}}, \underbrace{\mathtt{MKT}^{(t)}, \overbrace{\mathtt{RT}^{(t)}}^{M_{\mathsf{R}\to\mathsf{C}}^{(t,\mathsf{postA})}}, \mathtt{AD}^{(t)}}_{A_{\mathsf{R}}^{(t)}}, \underbrace{\mathtt{QUT}^{(t)}}_{M_{\mathsf{R}\leftarrow\mathsf{C}}^{(t,\mathsf{postA})}}).$$

The other two agents' observations, communications, and actions are defined similarly, with details provided in Section EC.1 of the e-companion. Table 1 summarizes these results.

3.3. Prompting LLM-Powered Agents: From Numbers to Text and Back

The system design optimization in (1) is numerical, with the design parameters θ and the system state ξ represented numerically.[2] In contrast, LLMs are text-native. They process inputs and produce outputs as sequences of tokens. A distinctive feature of LLM-MAS, compared to traditional simulation, is its *semantic interface*. This interface connects numerical data to text-based LLM queries through carefully constructed prompts, which facilitate embedding numbers into text and extracting them back out.

The data flow for simulating an LLM-powered agent's action is

$$\big(\underbrace{\theta}_{\text{design}}, \underbrace{H_i^{(t-1)}, O_i^{(t,\mathsf{preA})}, M_i^{(t,\mathsf{preA})}}_{\text{contextual information}}\big) \overset{\text{embedding}}{\longrightarrow} \underbrace{\mathcal{X}_i^{(t)} \to \text{LLM} \to \mathcal{Y}_i^{(t)}}_{\text{prompting}} \overset{\text{extraction}}{\longrightarrow} \underbrace{A_i^{(t)}}_{\text{action}}.$$

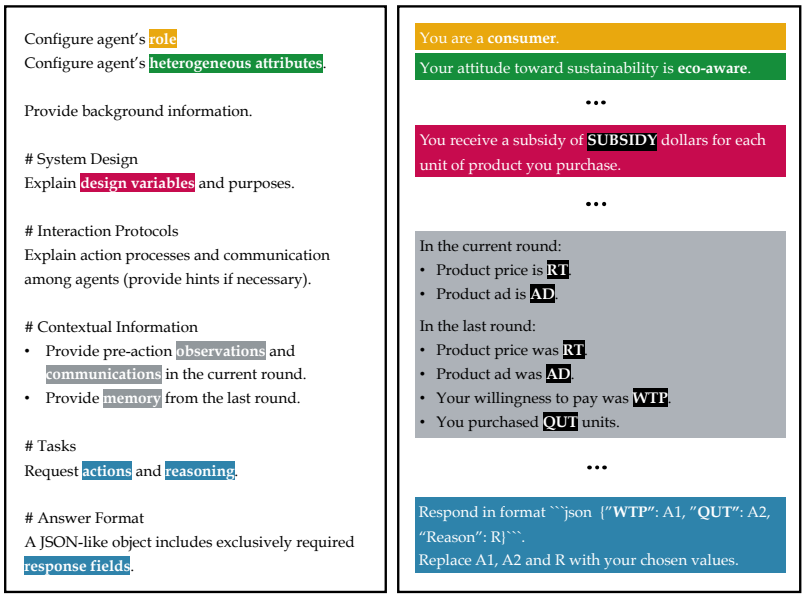
Here, the design and contextual information are embedded into the prompt $\mathcal{X}_i^{(t)}$. Querying the LLM with this prompt produces the textual output $\mathcal{Y}_i^{(t)}$, from which the agent's numerical action $A_i^{(t)}$ is extracted.

Both the embedding[3] and extraction steps make use of a prompt template (Figure 5). The left panel presents a generic template for prompting an LLM. It serves as a reference and can be adapted if alternative wording proves more effective in eliciting human-like behavior for a particular application. The right panel provides a concrete instance of the consumer agent from the supply chain example. The template has five primitive blocks, shown in different colors: agent's role (yellow), its attributes (green), system design (red), contextual information (gray), and output requirements (blue). Once defined for a specific agent, with the role $\mathcal{R}_i$ (e.g., "consumer") and attributes χ_i (e.g., "eco-aware"), the template remains *fixed* throughout LLM-MAS.

[2] We focus on problem instances where only the numerical components of ξ are involved in F.

[3] Here, "embedding" refers to populating prompt templates by replacing semantic variables (e.g., SUBSIDY) with specific numerical values (e.g., 1). This differs from "token embeddings", the high-dimensional representations of tokens in the latent space learned by LLMs during pre-training.

Figure 5 Prompt Template



Note. Left: General structure, highlighting the key information to be embedded. Right: Concrete instance for the consumer agent in Example 1. Ellipses represent additional details that help the agent interpret the highlighted content.

At run time, only the content inserted in the designated placeholders (black) varies. The content includes the design θ (e.g., `SUBSIDY`) and contextual information $(H_i^{(t-1)}, O_i^{(t,\text{preA})}, M_i^{(t,\text{preA})})$ relevant to the agent's action. For the consumer agent, no pre-action observation is available ($O_i^{(t,\text{preA})} = \varnothing$), the pre-action communication includes the retail price and the ad narratives communicated from the retailer ($M_i^{(t,\text{preA})} = (\texttt{RT}, \texttt{AD})$). The carried-over information from the previous round $H_i^{(t-1)}$ includes `RT`, `AD`, willingness to pay (`WTP`), and purchase quantity (`QUT`). See Table 1.

To simulate an agent's behavior, we prompt the LLM to generate specific actions and prescribe the output format in the prompt. A common choice is a structured `JSON` (e.g., ```` ```json {"WTP": A1, "QUT": A2, "Reason": R}``` ````). The desired output includes the action fields $A_i^{(t)}$ (e.g., `"WTP"` and `"QUT"`) and an auxiliary field (e.g., `"Reason"`) as a lightweight chain-of-thought reasoning for the chosen actions. We then parse the output string $\mathcal{Y}_i^{(t)}$ to extract `A1` and `A2` and convert them back to numbers.

4. Algorithm

A standard approach for solving the stochastic optimization problem (1) is to use first-order methods. For example, we can iteratively update the estimate of the optimal solution through

$$\theta_{k+1} = \Pi_\Theta(\theta_k - \eta_k \nabla f(\theta_k)), \tag{2}$$

where θ_k is the iterate at step k, $\eta_k > 0$ is the stepsize, and Π_Θ denotes the Euclidean projection onto the feasible set Θ. However, applying this method in the setting of LLM-MAS poses distinctive challenges. In the following, we first describe these challenges, then develop an algorithm to address them, and finally discuss variance reduction techniques to enhance the algorithm's efficiency.

4.1. Technical Challenges

The first challenge is gradient computation. Because the uncertainty is decision-dependent, we cannot calculate the gradient of $f(\theta) = \mathbb{E}_{\xi\sim\mu_\theta}[F(\theta;\xi)]$ by differentiating $F(\theta;\xi)$ alone. This issue is commonly resolved using a change of measure. Consider a probability distribution κ on Θ that is independent of θ and dominates μ_θ (i.e., μ_θ is absolutely continuous with respect to κ). Under regularity conditions that permit interchanging differentiation and expectation, one can show that

$$\nabla f(\theta) = \mathbb{E}_{\xi\sim\kappa}\Big[\underbrace{\nabla_\theta F(\theta;\xi)L(\theta;\xi) + F(\theta;\xi)\nabla_\theta L(\theta;\xi)}_{:=h(\theta;\xi)}\Big], \tag{3}$$

where $L(\theta;\xi) = (\mathrm{d}\mu_\theta/\mathrm{d}\kappa)(\xi)$ is the likelihood ratio (i.e., the Radon–Nikodym derivative) of μ_θ with respect to κ; see L'Ecuyer (1990). When (3) holds, $h(\theta;\xi)$ is an unbiased estimator of $\nabla f(\theta)$. Plugging it into (2) yields the stochastic gradient descent algorithm.

In LLM-MAS, however, $h(\theta;\xi)$ is intractable. Even though we may choose any θ-independent distribution κ, the likelihood ratio $L(\theta;\xi)$ is generally not computable because μ_θ is highly complex. Given θ, the generation of ξ is autoregressive, multi-agent, and multi-round: each token depends on the full prior context, and messages pass among agents over successive rounds, so small early random changes can cascade into large downstream shifts, creating deep dependencies. As noted in Remark 1, additional randomness arising from LLM quantization, floating-point arithmetic, and batching of concurrent requests further increases the complexity of μ_θ, rendering $L(\theta;\xi)$ practically intractable to evaluate. We address this challenge with a zeroth-order approach that estimates $\nabla f(\theta)$ from noisy function evaluations of $f(\theta)$ (with a controlled bias), avoiding the need to know or learn μ_θ.

The second challenge concerns steady-state performance. We cannot directly draw independent samples from μ_θ, the stationary distribution of the Markov chain. Instead, we can generate a long trajectory and use the terminal state $\xi^{(T)}$ to approximate μ_θ when T is large enough. LLM queries are costly because inference time scales with the number of tokens processed: the model first encodes the input tokens (prefill stage), then generates the output tokens one at a time (decode stage), conditioning on all previously seen tokens due to its autoregressive nature, and each token goes through the deep transformer layers. Consequently, long prompts and responses incur proportionally high computational costs. In the multi-round, multi-agent simulation, each round may need several queries per agent to generate actions and exchange messages, so token usage scales with both the time horizon and agent count. Therefore, simulating a sufficiently long trajectory is token-intensive and suffers from high latency. In the supply chain example, a 100-round simulation takes roughly half an hour on a local server with a small model (e.g., LLaMA3.1-8B).

The high cost of generating a long trajectory makes it prohibitive to use independent samples of $\xi^{(T)}$ for each iterate θ_k in the recursion (2). Instead of simulating many trajectories, we propose an algorithm that uses one single trajectory across optimization iterations to achieve substantial savings in LLM query cost.

4.2. Zeroth-Order Approach

For simplicity, assume for now that, given θ, we can sample ξ from the stationary distribution μ_θ (we relax this later). We estimate $\nabla f(\theta)$ by evaluating $F(\theta;\xi)$ at small perturbations of θ, with each ξ generated from the corresponding perturbed μ_θ. Many zeroth-order methods follow this framework; see LA and Bhatnagar (2025). We adopt a gradient estimator based on randomized, simultaneous perturbations (Nesterov and Spokoiny 2017), which is known for its ease of use and sample efficiency in multi-dimensional settings.

Consider a smoothed counterpart of the objective function $f(\theta)$:

$$f_\delta(\theta) := \mathbb{E}_{u\sim\Lambda}[f(\theta+\delta u)], \tag{4}$$

where $\delta > 0$ is the smoothing parameter (i.e., perturbation radius) and $u \in \mathbb{R}^d$ is a zero-mean random direction drawn from a d-dimensional distribution Λ. When δ is small, $f_\delta(\theta)$ and its gradient $\nabla f_\delta(\theta)$ closely approximate $f(\theta)$ and $\nabla f(\theta)$, respectively. We take $\Lambda = N(0, d^{-1}I_d)$, the standard multivariate Gaussian distribution normalized by the dimension d, which ensures $\mathbb{E}_{u\sim\Lambda}[\|u\|^2] = 1$. By Stein's identity,

$$\nabla f_\delta(\theta) = \frac{d}{\delta}\,\mathbb{E}_{u\sim\Lambda}[f(\theta+\delta u)u], \tag{5}$$

which leads to the unbiased estimator of $\nabla f_\delta(\delta)$:

$$G(\theta;\xi^\pm,u,\delta) := \frac{d}{2\delta}\left(F(\theta+\delta u;\xi^+) - F(\theta-\delta u;\xi^-)\right)u, \quad u\sim N(0,d^{-1}I_d), \tag{6}$$

where $\xi^\pm = (\xi^+,\xi^-)$, drawn from the perturbed distributions $\mu_{\theta+\delta u}$ and $\mu_{\theta-\delta u}$, respectively.

This estimator uses two samples of ξ associated with a pair of symmetric, random perturbations of θ, which is query-efficient for LLM-MAS. In contrast, the classic central finite-difference estimator applies coordinate-wise perturbations and needs two evaluations per dimension, resulting in a total of $\mathcal{O}(d)$ samples. This is often prohibitively costly in LLM-MAS when θ is multi-dimensional. Moreover, deterministic, coordinate-wise perturbations tend to under-explore the parameter space compared with randomized perturbations, potentially causing stagnation when optimizing nonconvex objectives (Chin 1997).

4.3. On-Trajectory Learning

To use the zeroth-order estimator (6) in the recursion (2), we face one main difficulty: we cannot sample directly from μ_θ. For any given θ, the stationary distribution of the Markov chain Ψ_θ can only be approximated through sufficiently long simulations of $\xi^{(t)}$.

A straightforward implementation then proceeds as follows. At each iterate θ_k, we simulate two trajectories, corresponding to the perturbations $\theta_k + \delta_k u_k$ and $\theta_k - \delta_k u_k$, where δ_k is the smoothing parameter and u_k is the random perturbation direction. The terminal state of each trajectory approximates a sample from the corresponding perturbed stationary distribution. We refer to this approach as the *multi-trajectory learning* (MTL) algorithm (Figure 6, right), because it requires many trajectories throughout the optimization.

Figure 6 On-Trajectory vs. Multi-Trajectory Learning

Algorithm 1 On-Trajectory Learning

Input: $\delta_0, \eta_0, \alpha, \beta, \xi_0, \theta_0$

Output: Last iterate θ_k

for $k = 0, 1, \ldots$ **do** ▷ LLM-MAS
  Set $\delta_k = \delta_0/(1+k)^\alpha$, $\eta_k = \eta_0/(1+k)^\beta$
  Draw $u_k \sim N(0, d^{-1} I_d)$
  Generate $\xi_k^+ = \Psi(\xi_k | \theta_k + \delta_k u_k)$
  Generate $\xi_k^- = \Psi(\xi_k | \theta_k - \delta_k u_k)$
  Compute $G_k = G(\theta_k; (\xi_k^+, \xi_k^-), u_k, \delta_k)$
  Set $\theta_{k+1} = \Pi_\Theta(\theta_k - \eta_k G_k)$
  Generate $\xi_{k+1} = \Psi(\xi_k | \theta_k)$
end for

Algorithm 2 Multi-Trajectory Learning

Input: $\delta_0, \eta_0, \alpha, \beta, T, \xi_0^{(0)}, \theta_0$

Output: Last iterate θ_k

for $k = 0, 1, \ldots$ **do**
  Set $\delta_k = \delta_0/(1+k)^\alpha$, $\eta_k = \eta_0/(1+k)^\beta$
  Draw $u_k \sim N(0, d^{-1} I_d)$
  for $t = 0, \ldots, T-1$ **do** ▷ LLM-MAS
    Generate $\xi_k^{+,(t+1)} = \Psi(\xi_k^{+,(t)} | \theta_k + \delta_k u_k)$
    Generate $\xi_k^{-,(t+1)} = \Psi(\xi_k^{-,(t)} | \theta_k - \delta_k u_k)$
  end for
  Compute $G_k = G(\theta_k; (\xi_k^{+,(T)}, \xi_k^{-,(T)}), u_k, \delta_k)$
  Set $\theta_{k+1} = \Pi_\Theta(\theta_k - \eta_k G_k)$
end for

Note. (i) For on-trajectory learning, the round index t used in LLM-MAS is dropped because it coincides with the algorithm iterate index k. (ii) For multi-trajectory learning, the trajectory in the inner loop over t is initialized as $\xi_k^{+,(0)} = \xi_k^{-,(0)} = \xi_0^{(0)}$ for all k.

However, generating long trajectories between successive iterations is costly in LLM-MAS. A single round of simulation can take tens of seconds or even minutes, and the runtime grows with the number of agents and communication links. Waiting for LLM-MAS to finish a long trajectory before updating design parameters is therefore excessively expensive. In addition, only the terminal state $\xi^{(T)}$ contributes to gradient estimation, while intermediate states $\xi^{(1)}, \ldots, \xi^{(T-1)}$ are discarded even though they potentially contain valuable information. This practice wastes LLM queries because each state transition consumes tokens, leading to high token utilization and long runtimes while providing limited information gain per query.

To address the sample inefficiency of MTL, which stems from decoupling parameter updates from gradient estimation, we propose an *on-trajectory learning* (OTL) algorithm (Figure 6, left). OTL operates on one single trajectory, along which it updates the design parameters and concurrently forms the zeroth-order gradient estimate using a pair of one-round simulations initialized from the same current state but executed under two perturbed designs. Specifically, at each iteration k,

$$\text{(Random Perturbation)} \qquad \begin{cases} u_k \sim N(0, d^{-1} I_d), \\ \xi_k^+ = \Psi(\xi_k | \theta_k + \delta_k u_k), \\ \xi_k^- = \Psi(\xi_k | \theta_k - \delta_k u_k), \end{cases} \tag{7}$$

$$\text{(Gradient Estimation)} \qquad G_k = G(\theta_k; \xi_k^\pm, u_k, \delta_k), \tag{8}$$

$$\text{(Parameter Update)} \qquad \theta_{k+1} = \Pi_\Theta(\theta_k - \eta_k G_k), \tag{9}$$

$$\text{(State Update)} \qquad \xi_{k+1} = \Psi(\xi_k|\theta_k), \tag{10}$$

where we write $\Psi_\theta(\xi)$ as $\Psi(\xi|\theta)$ for notational convenience.

The OTL trajectory does not come from the Markov chain Ψ_θ with a fixed θ; instead, each state is generated from the previous one under an adaptively changing θ in (10). The samples $\xi_k^\pm = (\xi_k^+, \xi_k^-)$ from random perturbations in (7) are generated under the transition distribution of the Markov chain with the perturbed design parameter, rather than under the corresponding stationary distributions assumed by the zeroth-order estimator (6). This mismatch introduces bias in estimating $\nabla f_{\delta_k}(\theta_k)$, the smoothed counterpart of the gradient $\nabla f(\theta_k)$.

To prevent the bias from hindering convergence of the iterates θ_k, we adopt a *two-timescale* scheme. The gradient estimate G_k evolves on the fast timescale, allowing it to track the targets induced by the current design parameter. The parameter estimate θ_k evolves on the slow timescale, updating only after the gradient estimate has effectively attained a nearly stationary consistent with θ_k. This two-timescale scheme is realized by choosing the perturbation radius δ_k for gradient estimation and the stepsize η_k for parameter updates such that both decay with iterates k, while satisfying $\eta_k/\delta_k \to 0$ as $k \to \infty$. A typical choice is $\delta_k = \delta_0/(1+k)^\alpha$ and $\eta_k = \eta_0/(1+k)^\beta$, where $\delta_0, \eta_0 > 0$ and $\alpha < \beta$; see Assumption 8 for details.

4.4. Asymptotic Convergence

As the objective function $f(\theta)$ in (1) is generally nonconvex, we establish convergence to a stationary point under the following assumptions, which are standard in the SA literature.

ASSUMPTION 1 **(Feasible Set)**. *Θ is a compact and convex subset of $\mathbb{R}^d$.*

ASSUMPTION 2 **(Objective Function)**. *$f(\theta)$ is continuously differentiable, and there exists a constant $L_f > 0$ such that $\|\nabla f(\theta) - \nabla f(\theta')\| \le L_f\|\theta - \theta'\|$ for all $\theta, \theta' \in \Theta$.*

ASSUMPTION 3 **(Random Function)**. *There exist constants $L_F > 0$ and $C_F > 0$ such that $|F(\theta;\xi) - F(\theta';\xi)| \le L_F\|\theta - \theta'\|$ for all $\theta, \theta' \in \Theta$ and all $\xi \in \Xi$, and $|F(\theta;\xi)| \le C_F$ for all $\theta \in \Theta$ and all $\xi \in \Xi$.*

ASSUMPTION 4 **(Transition Kernel)**. *For each $\theta \in \Theta$, let P_θ denote the Markov chain's transition kernel. There exists a constant $L_\mathsf{P} > 0$ such that $\mathcal{W}_1(\mathsf{P}_\theta(\xi,\cdot), \mathsf{P}_{\theta'}(\xi,\cdot)) \le L_\mathsf{P}\|\theta - \theta'\|$ for all $\theta, \theta' \in \Theta$ and all $\xi \in \Xi$, where $\mathcal{W}_1(\nu,\nu') = \inf_{\pi\sim\Gamma(\nu,\nu')} \mathbb{E}_{\zeta,\zeta'\sim\pi}[\|\zeta - \zeta'\|_1]$ is 1-Wasserstein distance with $\Gamma(\nu,\nu')$ denoting the set of all couplings of probability measures ν and ν'.*

ASSUMPTION 5 **(Geometric Mixing)**. *For each $\theta \in \Theta$, P_θ admits a unique stationary distribution μ_θ. Moreover, there exist constants $C_M > 0$ and $\gamma \in (0,1)$ such that $\|\mathsf{P}_\theta^{(k)}(\xi,\cdot) - \mu_\theta(\cdot)\|_{\mathrm{TV}} \le C_M\gamma^k$ for all $k \ge 0$ and all $\xi \in \Xi$, where $\mathsf{P}_\theta^{(k)}(\xi,\cdot) = \mathbb{P}(\xi^{(k)} \in \cdot|\xi^{(0)} = \xi;\theta)$ is the k-step transition distribution from the initial state ξ, and $\|\nu - \nu'\|_{\mathrm{TV}} = 1/2\int|\nu(\zeta) - \nu'(\zeta)|\,\mathrm{d}\zeta$ is the total variation distance between probability measures ν and ν'.*

ASSUMPTION 6 **(Transient Gradient)**. *For each $\theta \in \Theta$ and all $\xi \in \Xi$, there exists a function $H(\theta;\xi)$ such that $\nabla_\theta \int_\Xi F(\theta;\xi')\mathsf{P}_\theta(\xi,\mathrm{d}\xi') = \int_\Xi H(\theta;\xi')\mathsf{P}_\theta(\xi,\mathrm{d}\xi')$.*

Under Assumption 6, we have $\nabla f(\theta) = \mathbb{E}_{\xi\sim\mu_\theta}[H(\theta;\xi)]$ by the invariance of μ_θ. For any given θ, this yields the Poisson equation:

$$H(\theta;\xi) - \nabla f(\theta) = \nu_\theta(\xi) - \int_\Xi \nu_\theta(\xi')\mathsf{P}_\theta(\xi,\mathrm{d}\xi'). \tag{11}$$

Assumption 5 ensures the existence of a solution $\nu_\theta(\xi)$; see Meyn and Tweedie (2012, Chapter 17).

ASSUMPTION 7 **(Poisson's Equation)**. *There exist constants $L_\nu > 0$ and $C_H > 0$ such that*

(i) $\sup_{\theta\in\Theta} \|\nu_\theta(\xi) - \nu_\theta(\xi')\| \le L_\nu\|\xi - \xi'\|$ *for all $\xi, \xi' \in \Xi$;*

(ii) $\sup_{\theta\in\Theta} \|\nu_\theta(\xi)\| \le L_\nu(1 + \|\xi\|)$ *for all $\xi \in \Xi$;*

(iii) $\|\nu_\theta(\xi) - \nu_{\theta'}(\xi)\| \le L_\nu\|\theta - \theta'\|(1 + \|\xi\|)$ *for all $\theta, \theta' \in \Theta$ and all $\xi \in \Xi$; and*

(iv) $\mathbb{E}_{\xi'\sim\mathsf{P}_\theta(\xi,\cdot)}[\|H(\theta;\xi')\|^2] \le C_H$ *for all $\theta \in \Theta$ and all $\xi \in \Xi$.*

ASSUMPTION 8 **(Stepsizes)**. *$\eta_k \to 0$ and $\delta_k \to 0$ as $k \to \infty$; $\sum_{k=0}^\infty \eta_k = \infty$, $\sum_{k=0}^\infty \eta_k\delta_k < \infty$, and $\sum_{k=0}^\infty \eta_k^2/\delta_k^2 < \infty$.*

THEOREM 1. *Under Assumptions 1–8, the OTL iterates $\{\theta_k : k \ge 0\}$ converge almost surely to the set of stationary points $\Theta_0 = \{\theta \in \Theta : 0 \in \nabla f(\theta) + \mathcal{N}_\Theta(\theta)\}$, where $\mathcal{N}_\Theta(\theta) := \{x \in \mathbb{R}^d : \langle x, \vartheta - \theta\rangle \le 0, \forall \vartheta \in \Theta\}$ denotes the normal cone of Θ at θ.*

We analyze the OTL algorithm using the ordinary differential equation (ODE) approach for SA (Kushner and Yin 2003). We treat the recursions in (7)–(10) as a discretization of the limiting ODE and show that the iterates asymptotically track it, which yields convergence to a stationary point of problem (1).

Applying this classic approach to our setting raises technical complications due to intertemporal dependencies from adaptive Markovian data. Existing work on SA with such data typically assumes access to gradients, which distinguishes our setting. For example, Che et al. (2026) give a finite-time analysis of stochastic gradient descent based on Lyapunov functions and characterize conditions under which the convergence rate approaches that of the independent-noise case for objectives with structural properties such as convexity. Li et al. (2026) further refine the rate using the Poisson equation approach. L'Ecuyer and Glynn (1994) is most closely related to ours, which applies zeroth-order gradient estimation to steady-state stochastic optimization, but their analysis is limited to single-server queues and relies on the queueing structure.

We also follow the Poisson equation approach. The key in our proof is to control the bias that the zeroth-order gradient estimator (6) accumulates along the trajectory. First, we address the nonstationary bias from the coupling of parameter updates and gradient estimates in steps (8)–(10) using the two-timescale scheme in Assumption 8. This ensures that the transient gradient $H(\theta;\xi)$ (see Assumption 6) consistently aligns with the target $\nabla f(\theta)$. Second, we control the bias resulting from the zeroth-order approach in step (7) by exploiting the smoothness of the transition kernel under Assumption 4. This yields a bound on the gap between the gradient estimator (6) and $H(\theta;\xi)$. See Section EC.3 of the e-companion for details.

5. Efficiency Enhancement: Variance Reduction

Gradient estimation in Markov chains becomes increasingly challenging as the trajectory length grows because the variance of such estimators often rises sharply with time horizon (Asmussen and Glynn 2007, Chapter VII). In LLM-MAS, this difficulty is further amplified by multi-agent interactions within each simulation round. As agents adapt to one another, small local changes can spread through the system and lead to pronounced variability, making variance reduction crucial for efficient use of LLM queries.

In the following, we introduce two variance reduction techniques from the zeroth-order optimization literature and demonstrate how to incorporate them in LLM-MAS. They are complementary. The first applies when F depends on θ explicitly (i.e., $\nabla_\theta F(\theta;\xi) \neq 0$); it leverages the explicit coupling between the design parameters and system performance. The second has broader scope and covers cases where F depends on θ only through ξ (i.e., $\nabla_\theta F(\theta;\xi) = 0$); it instead uses the information accumulated during optimization to handle settings where the coupling between θ and F is implicit.

5.1. Guided Perturbation

In the gradient estimator (6), the perturbation vector u is drawn from an isotropic Gaussian, which explores all directions equally, whereas ideally one would follow the direction of the unknown gradient $\nabla f(\theta)$. The *guided perturbation* (GP) method (Maheswaranathan et al. 2019), initially developed for cases where no noise or only independent noise is present, addresses this by using a *surrogate gradient* ϕ to direct the perturbations toward more promising regions.

Specifically, the GP method samples u from a non-isotropic Gaussian $N(0,\Sigma)$, where the covariance matrix Σ is a weighted combination of an isotropic component and a gradient-aligned component:

$$\Sigma := w\frac{I_d}{d} + (1-w)\frac{\phi\phi^\intercal}{\|\phi\|^2}.$$

It stretches the spherical Gaussian into an ellipsoid elongated along the surrogate gradient direction. This structure features a distinct eigenvalue $w/d + (1-w)$ in the direction of ϕ, and $d-1$ identical eigenvalues w/d along the directions orthogonal to it. The weight $w \in [0,1]$ controls the trade-off between exploration and exploitation: a larger value promotes isotropic exploration, while a smaller value steers perturbations toward the surrogate gradient. When $w = 1$, the perturbation distribution reduces to $N(0, d^{-1}I_d)$ as in the basic gradient estimator (6). As w decreases, the perturbations become increasingly aligned with ϕ, concentrating the search along that direction while reducing the variability in orthogonal directions.

The key to applying this method is the choice of ϕ: it is problem-specific and should align closely with $\nabla f(\theta)$. In our setting, when θ enters F explicitly, $\nabla_\theta F(\theta;\xi)$ in (3) is nonzero. Although $\nabla_\theta F(\theta;\xi)$ is not an unbiased estimator of $\nabla f(\theta)$ due to the decision-dependent uncertainty, it remains correlated with the true gradient and is therefore informative. We set $\phi = \nabla_\theta F(\theta;\xi)$ to exploit this information in LLM-MAS. Formally, let the GP gradient estimator be denoted by

$$G^{\mathrm{GP}}(\theta;\xi,\xi^{\pm},u,\delta) = \frac{1}{2\delta}\left(F(\theta+\delta u;\xi^+) - F(\theta-\delta u;\xi^-)\right)u, \quad u \sim N(0,\Sigma(\xi)), \tag{12}$$

where we write $\Sigma(\xi)$ to highlight its dependence on ξ through the surrogate gradient ϕ.

We incorporate GP into OTL as follows. At each iteration k, we modify the first two steps (7) and (8) to

$$\text{(Random Perturbation)} \qquad \begin{cases} u_k \sim N(0, \Sigma(\xi_k)), \\ \xi_k^+ = \Psi(\xi_k | \theta_k + \delta_k u_k), \\ \xi_k^- = \Psi(\xi_k | \theta_k - \delta_k u_k), \end{cases} \tag{13}$$

$$\text{(Gradient Estimation)} \qquad G_k = G^{\mathrm{GP}}(\theta_k; \xi_k, \xi_k^{\pm}, u_k, \delta_k), \tag{14}$$

while the remaining steps follow those in (9) and (10).

Because Σ is positive definite, the expected value of this gradient estimator points in a valid direction. However, with non-isotropic perturbations drawn from $\Lambda = N(0, \Sigma)$, $G^{\mathrm{GP}}(\theta; \xi, \xi^{\pm}, u, \delta)$ is no longer unbiased with respect to the gradient of the smoothed objective $f_\delta(\theta) = \mathbb{E}_{u \sim \Lambda}[f(\theta + \delta u)]$ in (4); see Lemma EC.1 in Section EC.3.1 of the e-companion. To avoid this bias, arising from the dependence of $\Sigma(\xi)$ on ξ, that could harm the convergence of the OTL algorithm, we vary the weight w across iterations so that the GP gradient estimator asymptotically approaches the basic gradient estimator (6). Specifically, we set $w_{k+1} = 1 - \rho(1 - w_k)$ for some $\rho \in (0, 1)$, then $1 - w_k$ decays geometrically with contraction factor $1 - \rho$, implying that $w_k \to 1$. This scheduling enables guided exploration in early iterations while preserving the overall convergence, particularly when the smoothing parameter decays polynomially as $\delta_k = \delta_0/(k+1)^\alpha$.

5.2. Residual Feedback

When $\nabla_\theta F(\theta; \xi) = 0$, the system performance depends on θ only through the distribution of ξ, so the GP method does not apply. We instead use the *residual feedback* (RF) method (Zhang et al. 2022), originally proposed for deterministic or stochastic settings with independent noise, and extend it to our setting where the distribution of ξ_k changes adaptively with the iterate θ_k.

Unlike the basic gradient estimator (6) and GP gradient estimator (12), both of which use only samples in the current iteration, the RF method additionally uses samples from the previous iteration. If the noise were independent, denoted by ε_k to distinguish it from ξ_k, we would estimate $\nabla f(\theta_k)$ using one sample from iteration k and another from iteration $k-1$, via $d\delta_k^{-1}(F(\theta_k + \delta_k u_k; \varepsilon_k) - F(\theta_{k-1} + \delta_{k-1} u_{k-1}; \varepsilon_{k-1}))u_k$, where u_k and u_{k-1} are drawn from $N(0, d^{-1} I_d)$ as in (6). The difference between the two function evaluations represents the residual, as it carries information from the most recent evaluation and thus gives the method its name. It is conceptually related to the control variate method, which leverages correlations between the target quantity and suitably chosen auxiliary variables to reduce variance (Asmussen and Glynn 2007, Chapter V).

However, in our setting the "noise" ξ_k is Markovian, with the transition distribution changing adaptively with the iterate θ_k. Because θ_k itself is perturbed, it is necessary to specify the parameter value that governs the transition generating each sample; that is, the random variables are to replace ε_k and ε_{k-1}. We therefore define the RF gradient estimator as follows:

$$G^{\mathrm{RF}}(\theta_k; u_k, \delta_k, \xi_k^+, \theta_{k-1}, u_{k-1}, \delta_{k-1}, \xi_{k-1}^+)$$

$$:= \frac{d}{\delta_k} \left(F(\theta_k + \delta_k u_k; \xi_k^+) - F(\theta_{k-1} + \delta_{k-1} u_{k-1}; \xi_{k-1}^+) \right) u_k, \quad u_k, u_{k-1} \sim N(0, d^{-1} I_d), \tag{15}$$

where ξ_k^+ and ξ_{k-1}^+ are drawn from the perturbed distributions associated with parameters $\theta_k + \delta_k u_k$ and $\theta_{k-1} + \delta_{k-1} u_{k-1}$, respectively. Note that the second term in (15) has mean zero; that is, $\mathbb{E}[F(\theta_{k-1} + \delta_{k-1} u_{k-1}; \xi_{k-1}^+) u_k] = 0$ by the independence and zero mean of u_k. Therefore, $d\delta^{-1} F(\theta_{k-1} + \delta_{k-1} u_{k-1}; \xi_{k-1}^+) u_k$ serves as a control variate that pairs with the first term $d\delta^{-1} F(\theta_k + \delta_k u_k; \xi_k^+) u_k$, which is an unbiased estimator of $\nabla f_{\delta_k}(\theta_k)$ by (5), to reduce its variance.

The RF gradient estimator mirrors the structure of the basic gradient estimator $G(\theta; \xi^{\pm}, u, \delta)$ yet requires only one new function evaluation per iteration, reducing the computational cost (LLM queries) by half. Rather than explicitly performing the second function evaluation as in (6), this estimator implicitly reconstructs it using past data. For further comparison, consider a variant $\widetilde{G}(\theta; \xi^+, u, \delta) := d\delta^{-1} F(\theta + \delta u; \xi^+) u$ that incurs the same computational cost as the RF gradient estimator and is also unbiased for $\nabla f_\delta(\theta)$ when $u \sim N(0, d^{-1} I_d)$. However, this estimator has substantially higher variance because it lacks a control variate.

To incorporate RF into OTL, we modify the first two steps (7) and (8) of each iteration k to

$$\text{(Random Perturbation)} \qquad \begin{cases} u_k \sim N(0, d^{-1} I_d), \\ \xi_k^+ = \Psi(\xi_k | \theta_k + \delta_k u_k), \end{cases} \tag{16}$$

$$\text{(Gradient Estimation)} \qquad G_k = G^{\mathrm{RF}}(\theta_k; u_k, \delta_k, \xi_k^+, \theta_{k-1}, u_{k-1}, \delta_{k-1}, \xi_{k-1}^+), \tag{17}$$

while the remaining steps follow those in (9) and (10).

When F varies smoothly as the design parameters move from $\theta_{k-1} + \delta_{k-1} u_{k-1}$ to $\theta_k + \delta_k u_k$, the evaluations become correlated. This correlation allows the second term in (15) to act as a control variate, effectively offsetting part of the Monte Carlo noise in the gradient estimator. As a result, the RF method retrieves the information over the course of the algorithm's learning process, thereby achieving variance reduction.

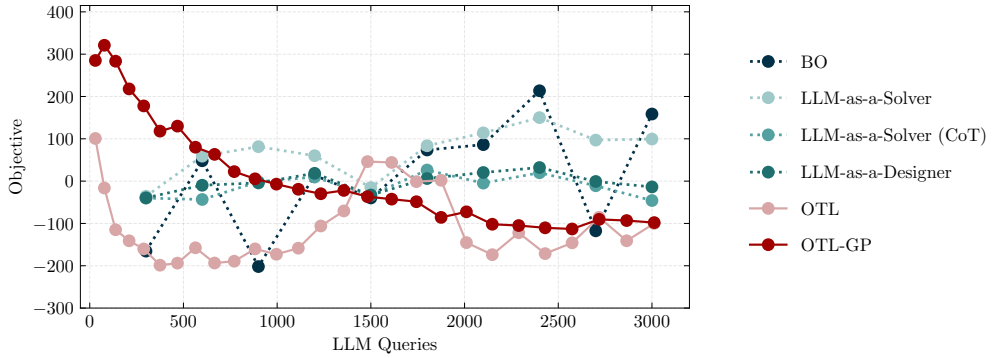
6. Numerical Experiments: Efficiency Comparison

We evaluate our algorithm on the supply chain example. The policymaker chooses a carbon tax θ_1 and a consumer subsidy θ_2 to minimize the objective $F(\theta; \xi)$ that balances supply chain welfare, fiscal costs, and environmental externalities. We also incorporate agent-specific attributes to reflect their preferences regarding sustainability initiatives. Additional details are provided in Section EC.1 of the e-companion.

We implement the OTL algorithm and enhance it with the GP method (OTL-GP) since $\nabla_\theta F(\theta; \xi)$ is nonzero in this example. We also consider the following benchmarks.

(i) Bayesian optimization (BO): A global optimization method that treats the objective $f(\theta)$ as a black-box and builds a surrogate (typically a Gaussian process) to guide the search for optimal solutions without gradient information (Frazier 2018). Because BO requires independent samples of ξ from the steady-state distribution μ_θ, each iteration in our implementation uses the terminal state $\xi^{(T)}$ of a long trajectory ($T = 100$) to approximate μ_θ, as in the MTL algorithm.

Figure 7 Algorithm Performance



Note. (i) LLaMA3.1-8B is the backbone for all agents. (ii) Each one-round simulation operator Ψ_θ requires three LLM queries. The additional query for mimicking the retailer agent's tool-call (for ad-narrative generation) is not included in the count.

(ii) LLM-as-a-Solver: An LLM is prompted to perform numerical optimization without domain context. Solutions and their objective values from previous iterations are provided in each subsequent prompt to request new candidates. In this way, the LLM refines its search over iterations, analogous to conventional numerical optimization methods, and, like BO, explores multiple trajectories.

(iii) LLM-as-a-Solver (CoT): Similar to the above, but applies the chain-of-thought prompting technique (Wei et al. 2022) to strengthen reasoning and help the LLM refine later solutions.

(iv) LLM-as-a-Designer: An LLM acts as the system designer and proposes solutions using its internal logic, without being given the stochastic optimization formulation in (1). The prompt includes the actions of all agents simulated via LLM-MAS and the corresponding system performance from each round. This information guides the LLM to propose improved designs over successive iterations, using a single trajectory as in the OTL algorithm.

When comparing these algorithms, we track how the objective changes with the number of LLM queries, not the number of iterations, because some use one trajectory per iteration while others use a single trajectory across all iterations. The results are shown in Figure 7.

OTL reduces the objective quickly within the first 500 LLM queries. It then rebounds, declines again, and stabilizes around 2000 queries. This pattern indicates high sensitivity to noise in LLM-MAS, even though OTL eventually converges. In contrast, OTL-GP benefits from the variance reduction technique and shows a steady decrease without significant oscillations. Its learning curve also drops early, though more slowly than OTL, and it improves the solution consistently over time.

BO updates less frequently than OTL and OTL-GP because it relies on an independent long trajectory for each sample. Within the first 500 LLM queries, it occasionally reaches low objective values, but the learning curve then fluctuates widely and remains unstable even after 2000 queries, with large values indicative of continued exploration. This behavior arises mainly because the high query cost of independent samples limits the number of iterations, which slows the surrogate model in learning the objective function and delays

exploitation, especially in the presence of heavy noise. Exploration is critical for the convergence of BO to a globally optimal solution, but in this setting with limited query budget and noisy samples, much of the benefit is lost, leading to unsatisfactory performance.

When used purely as a numerical solver (LLM-as-a-Solver) without domain context, the LLM performs poorly, and incorporating CoT reasoning offers only marginal gains. The model acts as a language-based learner, drawing on previous observations to generate new candidate solutions. Unlike BO, it neither constructs a surrogate of the objective nor employs an acquisition function to balance exploration and exploitation. Instead, the LLM relies on heuristics and infers patterns from text without a careful treatment of uncertainty. Consequently, its ability to systematically explore the solution space is restricted, resulting in substantial performance degradation.

Assigning the LLM to role-play as a policymaker (LLM-as-a-Designer) and providing sufficient application background still fails to yield notable gains. The prompt enriches domain understanding but does not endow the model with a principled search strategy, so it can only explore heuristically. Because the LLM operates on a single trajectory where the temporal dependence among system states is difficult to untangle, it struggles to judge whether a change in the solution improves or hurts the performance.

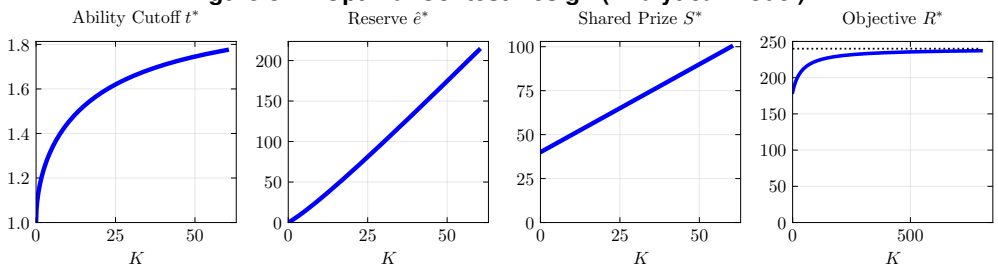
These results underscore the need to frame system design as a numerical optimization problem, rather than relying on plain-language descriptions of the problem and directly prompting an LLM for decisions. They also align with Simchi-Levi et al. (2026), who argue that LLMs should complement, not substitute, mathematically grounded models for optimizing complex stochastic systems. In line with this principle, our framework is best viewed as a human-in-the-loop approach: a human serves as the system designer, while LLM-MAS acts as an assistive component within the system.

7. Case Study: Innovation Contest Design

We apply the LLM-MAS framework to optimal contest design. Innovation contests are open calls for solutions to complex problems. Individuals or teams compete to submit the best ideas, models, or prototypes. In the digital economy, contests harness global expertise at scale, cut experimentation costs, and speed up discovery by turning hard tasks into merit-based competitions (Hu and Wang 2021, Nittala and Erat 2026). Platforms like InnoCentive, TopCoder, and Kaggle, along with landmark competitions such as the Netflix Prize, show how contests mobilize talent and disseminate advanced methods across industries. For a broader discussion of innovation platforms; see Chen et al. (2020).

Consider a contest designer seeking a mechanism that incentivizes contestants to exert as much effort as possible, thereby maximizing the expected total effort. We focus on a specific mechanism: a modified all-pay auction with an *entry fee* and a *reserve*, which has drawn considerable interest in the contest design literature. Under this mechanism, the designer sets a prize budget V. All contestants independently decide whether to enter the contest by paying an entry fee E and, conditional on entry, how much effort to exert. If

Figure 8 Optimal Contest Design (Analytical Model)



Note. Following Huang et al. (2026), we set $N = 3$, $V = 120$ and $t_0 = 0$, with t_i drawn from $\mathrm{Unif}[1, 2]$ for all i. The limiting value of R^* is $bV = 240$, extracting all surplus except that of the highest-ability contestant, which occurs with probability zero.

the highest effort exceeds a threshold (i.e., the reserve $\hat{e}$), the top performer receives the prize V and all entry fees, whereas the others receive nothing. Otherwise, if all participants' efforts fall below the reserve, each receives the same amount of a *shared prize* S that equals the entry fee plus a portion of V.

A salient feature of this mechanism is that the shared prize creates a chance of positive payoffs, which encourages low-ability contestants, who incur higher costs for any given effort level, to participate. At the same time, their entry fees are effectively transferred to the winner, raising the top performer's payoff beyond the base prize budget and thereby strengthening high-ability contestants' incentives to exert effort.

7.1. Analytical Model

This mechanism is characterized by a three-element tuple of design parameters $(E, \hat{e}, S)$. Under standard assumptions—including full rationality, risk neutrality, and independently and identically distributed abilities—as well as a common, publicly known *liability* K that caps each contestant's entry fee, Liu et al. (2018) use a game-theoretic analysis to derive in closed form the design parameters that maximize the expected total effort.[4] The optimal parameter values are expressed as functions of K. Specifically,

$$(E^*, \hat{e}^*, S^*) = \left(K,\, t^*(K) \left[(V + NK) F^{N-1}(t^*(K)) - K \right],\, \frac{K}{F^{N-1}(t^*(K))} \right), \tag{18}$$

where N is the number of contestants, F is the cumulative distribution function of a contestant's ability on a finite support $[a, b]$, and $t^*(K)$ is the cutoff separating low- and high-ability contestants. See Figure 8 for how the optimal design parameters and the maximum total effort R^* vary with K. The closed-form expressions of $t^*(K)$ and R^* are given in Section EC.2 of the e-companion.

Optimal Prize Allocation. If no contestant's effort exceeds $\hat{e}^*$, each receives the shared prize S^*. Otherwise, the contestant exerting the highest effort receives $V + NK$, which is the base prize plus the total entry fees, while all others receive 0. Since each participant pays the entry fee $E^* = K$, his net payoff becomes $-K$ (a "negative prize") whenever the highest effort exceeds $\hat{e}^*$ but he does not win.

[4] Huang et al. (2026) further show that the same contest structure also maximizes the expected winner's effort (when K is sufficiently large), though the optimal parameter values may differ from those that maximize the expected total effort. We focus on the total-effort objective; the analysis for the winner-effort objective proceeds in a similar manner.

Individual Behavior. Under this optimal design, all contestants choose to enter irrespective of their ability, but their efforts differ by ability. Low-ability contestants (those with $t_i \leq t^*(K)$) exert zero effort, whereas those with $t_i > t^*(K)$ exerts positive effort that increases with ability. See Section EC.2 of the e-companion for the closed-form expression of effort and Figure 9 for an illustration.

7.2. Behavioral Data from Lab Experiments

Huang et al. (2026) conducted lab experiments to evaluate the game-theoretic optimal designs described in Section 7.1. Six contest designs varying in the liability K and design objective were tested; see Table EC.1 in Section EC.2 of the e-companion. The experiments took place in March 2022 and April 2024 with 432 subjects recruited from a university, who were randomly assigned across six designs (72 per design). Under each design, 72 subjects were grouped into 24 contests, with $N = 3$ contestants per contest.

Before each contest, participants were informed of the design and their privately known ability, which was drawn from the interval $[1, 2]$ uniformly at random. They then decided simultaneously whether to enter. A contestant who chose not to enter made no further decisions. Otherwise, he paid the entry fee and chose an effort level. Because entry decisions were simultaneous, contestants did not know how many others would enter, so they did not know the exact prize amount until the session ended.

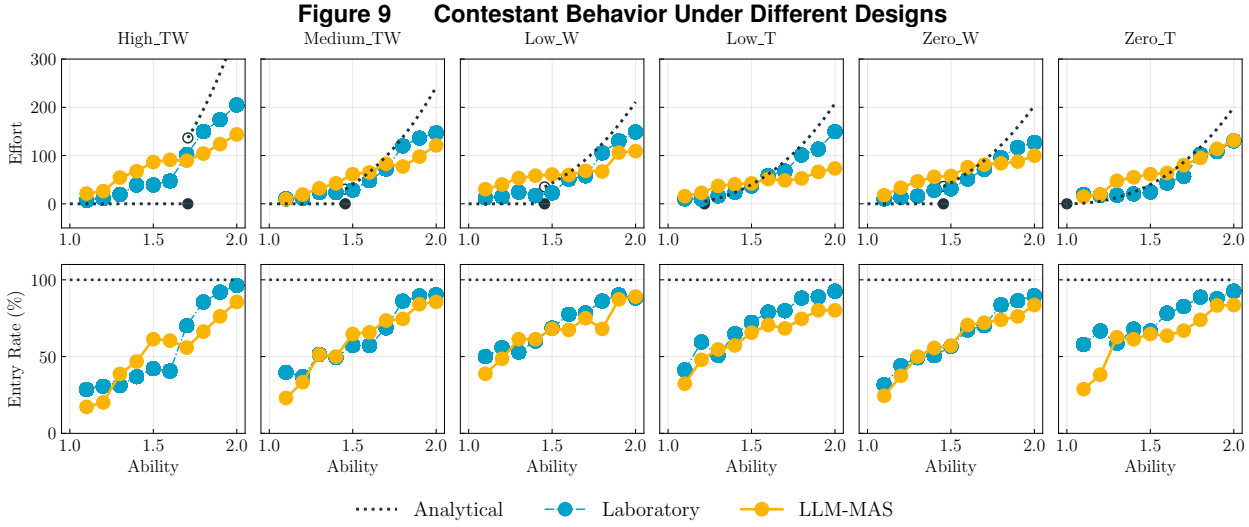
Across all designs, contestants' behavior differed markedly from game-theoretic predictions; see Figure 9. The theory predicts that (i) all contestants should enter regardless of ability, and (ii) low-ability contestants should exert zero effort, with a discontinuity in effort at the ability cutoff $t^*(K)$. In contrast, entry rates in the lab were well below 100%, rising gradually from nearly zero for low-ability contestants to nearly 100% for high-ability contestants. Moreover, low-ability contestants exerted positive effort, and effort increases smoothly with ability, showing no visible discontinuity.

After the experiments, a survey was administered to collect personal information from each participant, including demographics, risk attitudes, personality traits, and cognitive reflection test (CRT) score. The CRT score serves as a proxy for impulsive versus reflective decision-making and is widely used in experimental economics to study strategic behavior (Frederick 2005). Huang et al. (2026) used this information to explain the discrepancies between observed behavior and theoretical predictions. For example, while the theory assumes full rationality and risk neutrality, participants were generally risk-averse and exhibited loss aversion, which is a common form of bounded rationality.

7.3. LLM-MAS as a Design Evaluator

We replicate the configurations of Huang et al. (2026) to perform LLM-MAS, with each LLM-powered agent simulating a contestant. The agents are powered by gpt-oss-20B, an open-source LLM from OpenAI. The LLM prompts are adapted from the instructions given to lab participants, and each agent generates two actions: whether to enter and how much effort to exert. To reflect individual heterogeneity, each agent is endowed with a *persona* (i.e., attributes χ_i) randomly drawn from the pool of collected personal information.

Figure 9 Contestant Behavior Under Different Designs



Note. Ability cutoff t^* (left to right): 1.707, 1.455, 1.455, 1.218, 1.455 and 0.000.

It is four-dimensional: gender, risk tolerance (1 = "not risk-taking at all" to 7 = "extremely risk-taking"), competitiveness (1 = "not competitive at all" to 7 = "extremely competitive"), and CRT score (0 to 3).

The simulation results are shown in Figure 9. Across the six contest designs, LLM-powered agents' behavior closely matches that of human participants in the lab. As with humans, their effort increases with the endowed ability, and entry rates increase from almost zero for low-ability agents to nearly full participation for high-ability ones. This alignment is remarkable given that we did not fine-tune or post-train the model, and it was released months before the publication of Huang et al. (2026), so data leakage is not a concern.

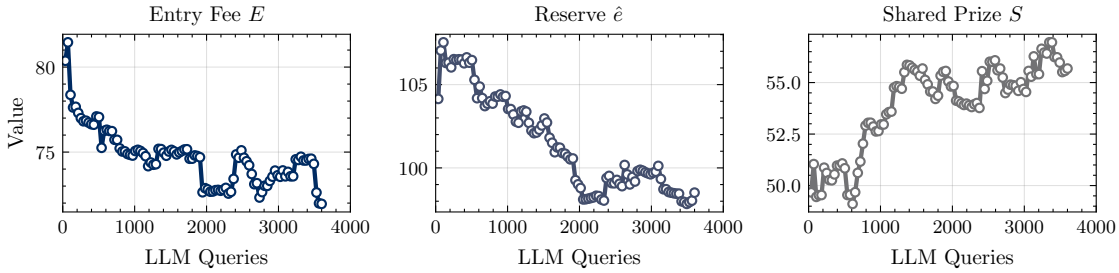
Because lab experiments are costly and time-consuming (the studies in Huang et al. (2026) took months, whereas an equivalent number of experiments can be completed in days with LLM-MAS), these results highlight the great potential of LLM-MAS as a cost-effective evaluator for system designs.

In addition to its cost efficiency, LLM-MAS offers another advantage: it can capture agents' "real-time" reasoning processes, providing valuable insights for performance analysis. In contrast, lab and field studies rarely observe such data, unless gathered through post-experiment interviews. In those cases, one can only infer the causes of deviations from theoretical predictions indirectly. The following examples show reasoning traces help explain why contestants' entry rate and effort mismatch the theory.

(i) Contrary to the predicted 100% entry rate, an LLM-powered agent (male; ability 1.613; risk tolerance 2; competitiveness 2; CRT score 3) chose not to enter. His stated reasoning "`Low risk tolerance leads to a conservative strategy of not entering to avoid potential loss from competition.`" suggests that low risk tolerance drives his conservative action. His inclination towards avoiding losses over chasing uncertain gains reflects his self-reported risk attitude and competitive disposition.

(ii) Contrary to the predicted zero effort from low-ability contestants, another LLM-powered agent (female; ability 1.287, below the cutoff of 1.802; risk tolerance 3; competitiveness 6; CRT score 2) chose to

Figure 10 Optimizing Contest Design Parameters



enter and invested 100 points out of her 300-point endowment. She anticipated one other would enter as well. Her stated reasoning "`I am moderately risk-averse but highly competitive, so I choose to enter with a modest high investment just above the reserve to maximize potential payoff while keeping cost reasonable.`" implies that high competitiveness counteracts risk aversion and leads to participation with a careful but proactive investment. This also illustrates a trade-off between risk exposure and potential returns.

7.4. LLM-MAS as a Design Explorer

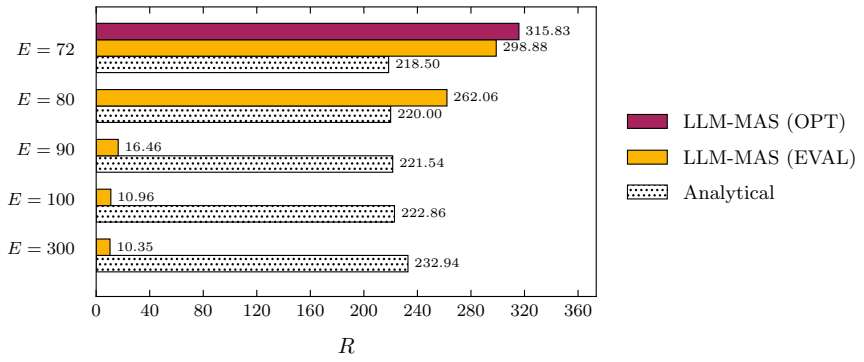
In the game-theoretic optimal design (18), the parameters are coupled through the liability K: once the entry fee is fixed, the reserve and shared prize are consequently determined. Using LLM-MAS, we can relax this coupling and optimize $\theta = (E, \hat{e}, S)$ as independent parameters over the feasible set $\Theta = [0, 300] \times [0, 1000] \times [0, 300]$. Consistent with the lab experiments, we set the upper bound of the entry fee to $K = 300$.

The random quantity ξ in the objective function corresponds to the vector of contestants' efforts, over which the expected total effort is computed. The contest is single-round, so ξ does not evolve over time. We can simulate this quantity in one round rather than running a long Markov chain to approximate the steady state, as in the supply chain example. We therefore apply the MTL algorithm to optimize θ. Here, the objective does not depend on θ explicitly, but only through the distribution of ξ. The GP method is thus not applicable, and we instead use the RF method for variance reduction. See Figure 10 for the iterates of the MTL-RF algorithm.

The resulting parameters are $E_{\text{OPT}} = 71.95$, $\hat{e}_{\text{OPT}} = 98.52$, and $S_{\text{OPT}} = 55.68$. The main difference from the game-theoretic design lies in the substantially lower entry fee. According to (18), the designer should set the entry fee as high as possible to maximize incentives by topping up the prize pool. With $K = 300$, this corresponds to an entry fee of 300 and an exceptionally high reserve of 1163.60. The expected total effort is then 232.94, as given by (EC.2.1) in Section EC.2 of the e-companion.

Under this high-fee, high-reserve design, the analytical model predicts that most contestants would exert zero effort, while only a few with abilities close to $b = 2$ (because the cutoff t^* is nearly 2) exert substantial effort above the reserve to have a chance of winning. However, LLM-MAS shows that with such high fees and reserves, even high-ability contestants opt out. They exert zero effort, like the others, to secure a high

Figure 11 Total Effort Under Contest Design Optimized via LLM-MAS



Note. (i) LLM-MAS (OPT): from LLM-MAS under the design identified by our algorithm. (ii) LLM-MAS (EVAL): from LLM-MAS under the game-theoretic design (18). (iii) Analytical: from the analytical model (EC.2.1) under the game-theoretic design (18).

chance of receiving the shared prize, which is safer and easier than expending extremely high effort to compete for the top prize. This, in turn, drives total effort near zero, which is precisely the opposite of what the designer intended. Notably, setting moderately high entry fees such as $E = 90$ or 100 reproduces this phenomenon via LLM-MAS as well.

In contrast to the near-zero effort observed under high-entry barriers, LLM-MAS estimates that the design $(E_{\text{OPT}}, \hat{e}_{\text{OPT}}, S_{\text{OPT}})$ identified by our algorithm yields a total effort of 315.83, which even exceeds the analytical model's maximum prediction with unlimited liability (Figure 8). See Figures 11 and 12 for comparisons across designs in total effort and contestant behavior, respectively.

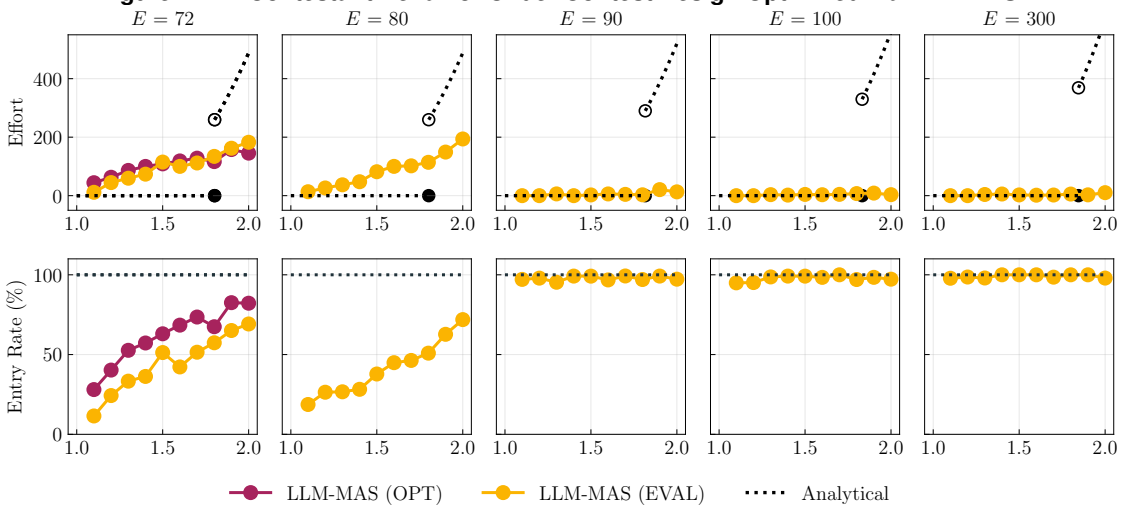
Furthermore, with the same entry fee $E_{\text{OPT}} = 71.95$, our optimal design features a much lower reserve and shared prize ($\hat{e}_{\text{OPT}} = 98.52$, $S_{\text{OPT}} = 55.68$) than the game-theoretic design ($\hat{e}_{\text{ANLY}} = 258.06$, $S_{\text{ANLY}} = 111.64$). As shown in Figure 12, this configuration induces higher entry rates across ability levels, because it reduces downside risk and makes entry appear safer to risk-averse contestants. While the per-entrant effort is unchanged, the larger pool of entrants raises total effort (315.83 versus 298.88; see Figure 11).

By simulating how people actually behave, this case study reveals the failure of charging entry fees up to the liability limit and shows that lower reserves and smaller shared prizes can boost participation without increasing individual effort, which lifts total effort for the designer. This highlights LLM-MAS as a design explorer: it allows researchers and practitioners to discover and test alternative designs that analytical models alone would miss.

8. Concluding Remarks

In this paper, we propose a principled framework for applying LLM-MAS to optimize service operations. Compared with traditional numerical simulation, LLM-MAS can generate and process rich information, including numeric, textual, and other types of unstructured data. This enables comprehensive representations of agent heterogeneity, and simulations of boundedly rational responses to system configurations and interactions with other agents. We model the uncertainty within LLM-MAS as a controlled Markov chain,

Figure 12 Contestant Behavior Under Contest Design Optimized via LLM-MAS



Note. (i) Ability cutoff t^* (left to right): 1.802, 1.816, 1.832, 1.845 and 1.939. (ii) LLM-MAS (OPT): from LLM-MAS under the design identified by our algorithm. (iii) LLM-MAS (EVAL): from LLM-MAS under the game-theoretic design (18). (iv) Analytical: from the analytical model under the game-theoretic design (18).

providing a theoretical basis for our OTL algorithm. This algorithm simultaneously updates the zeroth-order gradient estimates and design parameters on a single simulation trajectory, achieving significant savings in LLM queries. We further adapt two variance reduction techniques to address the high variability inherent in the long-horizon, multi-agent simulations.

We demonstrate the practical value of our LLM-MAS framework and algorithm through two applications. In a supply-chain application, OTL and its variant solve the optimization problem effectively, while the black-box method and LLMs used as numerical solvers or as role-playing system designers fail to produce meaningful results. This also suggests that LLMs are a complement, not a substitute, for OR/OM methods. A contest-design case study with real behavioral data shows that LLM-MAS can serve two roles: a cost-effective evaluator for testing known designs and an exploratory tool to uncover better designs that traditional methods may overlook. It thus complements both analytical models and empirical studies via lab or field experiments.

Several directions remain to enhance the practical value of LLM-MAS. For example, scaling LLM-MAS to large systems with many agents would necessitate tighter control of query costs without compromising fidelity. Promising approaches include surrogate modeling at both the agent and system levels, streamlined multi-agent communication, and advanced variance-reduction methods for long horizons and dense interactions. In addition, new algorithms are needed for high-dimensional design spaces, since zeroth-order methods degrade as problem dimension grows. Another direction is to extend LLM-MAS with a broader ecosystem of tools and multimodalities, so that it can simulate more complex human activities and handle images, audio, video, and graphs beyond text. This would create richer simulation environments and support applications such as simulating consumer reactions to various advertising creatives in digital platform operations.

References

Allon G, Cohen MC, Sinchaisri WP (2023) The impact of behavioral and economic drivers on gig economy workers. *Manufacturing & Service Operations Management* 25(4):1376–1393.

Allouah A, Besbes O, Figueroa JD, Kanoria Y, Kumar A (2025) What is your AI agent buying? Evaluation, implications and emerging questions for agentic e-commerce. https://arxiv.org/abs/2508.02630.

Argyle LP, Busby EC, Fulda N, Gubler JR, Rytting C, Wingate D (2023) Out of one, many: Using language models to simulate human samples. *Political Analysis* 31(3):337–351.

Asmussen S, Glynn PW (2007) *Stochastic Simulation: Algorithm and Analysis* (Springer).

Axtell RL, Farmer JD (2025) Agent-based modeling in economics and finance: Past, present, and future. *Journal of Economic Literature* 63(1):197–287.

Bandi N, Cohen MC, Ray S (2024) Behavioral retail operations: Tactics to win customers. *Foundations and Trends® in Technology, Information and Operations Management* 18(3-4):214–420.

Bertsimas D, Margaritis G (2024) Robust and adaptive optimization under a large language model lens. https://arxiv.org/abs/2501.00568.

Che E, Dong J, Tong XT (2026) Stochastic gradient descent with adaptive data. *Operations Research*, forthcoming.

Chen Y, Kirshner SN, Ovchinnikov A, Andiappan M, Jenkin T (2025) A manager and an AI walk into a bar: Does ChatGPT make biased decisions like we do? *Manufacturing & Service Operations Management* 27(2):354–368.

Chen Y, Liu TX, Shan Y, Zhong S (2023) The emergence of economic rationality of GPT. *Proceedings of the National Academy of Sciences* 120(51):e2316205120.

Chen YJ, Dai T, Korpeoglu CG, Körpeoğlu E, Sahin O, Tang CS, Xiao S (2020) OM forum—Innovative online platforms: Research opportunities. *Manufacturing & Service Operations Management* 22(3):430–445.

Chin DC (1997) Comparative study of stochastic algorithms for system optimization based on gradient approximations. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 27(2):244–249.

Cohen MC, Dai T, eds. (2026) *AI in Supply Chains: Perspectives from Global Thought Leaders* (Springer).

Cohen MC, Lobel R, Perakis G (2016) The impact of demand uncertainty on consumer subsidies for green technology adoption. *Management Science* 62(5):1235–1258.

Dai T, Swaminathan JM (2026) Artificial intelligence and operations: A foundational framework of emerging research and practice. *Production and Operations Management*, forthcoming.

Dao T, Fu D, Ermon S, Rudra A, Ré C (2022) FlashAttention: Fast and memory-efficient exact attention with IO-awareness. *Advances in Neural Information Processing Systems 35*, 16344–16359.

Davis AM (2018) Biases in individual decision-making. Donohue K, Katok E, Leider S, eds., *The Handbook of Behavioral Operations*, 149–198 (John Wiley & Sons).

Donohue K, Katok E, Leider S, eds. (2018) *The Handbook of Behavioral Operations* (John Wiley & Sons).

Donohue K, Özer O, Zheng Y (2020) Behavioral operations: Past, present, and future. *Manufacturing & Service Operations Management* 22(1):191–202.

Drusvyatskiy D, Xiao L (2023) Stochastic optimization with decision-dependent distributions. *Mathematics of Operations Research* 48(2):954–998.

Frazier PI (2018) Bayesian optimization. Gel E, Ntaimo L, eds., *Recent Advances in Optimization and Modeling of Contemporary Problems*, 255–278, INFORMS TutORials in Operations Research (INFORMS).

Frederick S (2005) Cognitive reflection and decision making. *Journal of Economic Perspectives* 19(4):25–42.

Gao C, Lan X, Li N, Yuan Y, Ding J, Zhou Z, Xu F, Li Y (2024) Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications* 11(1):1259.

Goli A, Singh A (2024) Frontiers: Can large language models capture human preferences? *Marketing Science* 43(4):709–722.

Hu M, Wang L (2021) Joint vs. separate crowdsourcing contests. *Management Science* 67(5):2711–2728.

Huang C, Tang Z, Hu S, Jiang R, Zheng X, Ge D, Wang B, Wang Z (2025) ORLM: A customizable framework in training large models for automated optimization modeling. *Operations Research* 73(6):2986–3009.

Huang L, Zhang J, Zhang J (2026) Optimal contests with negative prizes: Theory and experiment. *Management Science*, forthcoming.

Huang T, Allon G, Bassamboo A (2013) Bounded rationality in service systems. *Manufacturing & Service Operations Management* 15(2):263–279.

Jaillet P, Jiang J, Mellou K, Molinaro M, Podimata C, Zhou Z (2025) Online scheduling for LLM inference with KV cache constraints. `https://arxiv.org/abs/2502.07115`.

Konda VR, Tsitsiklis JN (2003) On actor-critic algorithms. *SIAM Journal on Control and Optimization* 42(4):1143–1166.

Krass D, Nedorezov T, Ovchinnikov A (2013) Environmental taxes and the choice of green technology. *Production and Operations Management* 22(5):1035–1055.

Kushner HJ, Yin GG (2003) *Stochastic Approximation and Recursive Algorithms and Applications* (Springer), 2nd edition.

LA P, Bhatnagar S (2025) Gradient-based algorithms for zeroth-order optimization. *Foundations and Trends in Optimization* 8(1-3):1–332.

L'Ecuyer P (1990) A unified view of the IPA, SF, and LR gradient estimation techniques. *Management Science* 36(11):1364–1383.

L'Ecuyer P, Glynn PW (1994) Stochastic optimization by simulation: Convergence proofs for the GI/G/1 queue in steady-state. *Management Science* 40(11):1562–1578.

Li X, Liang J, Chen X, Zhang Z (2026) Convergence and inference of stream stochastic gradient descent, with applications to queueing systems and inventory control. *Operations Research*, forthcoming.

Li Y, Dai J, Peng T (2025) Throughput-optimal scheduling algorithms for LLM inference and AI agents. https://arxiv.org/abs/2504.07347.

Liu B, Lu J, Wang R, Zhang J (2018) Optimal prize allocation in contests: The role of negative prizes. *Journal of Economic Theory* 175:291–317.

Maheswaranathan N, Metz L, Tucker G, Choi D, Sohl-Dickstein J (2019) Guided evolutionary strategies: Augmenting random search with surrogate gradients. *International Conference on Machine Learning*, 4264–4273.

Meyn SP, Tweedie RL (2012) *Markov Chains and Stochastic Stability* (Springer), 2nd edition.

Mitzenmacher M, Shahout R (2025) Queueing, predictions, and large language models: Challenges and open problems. *Stochastic Systems* 15(3):195–219.

Nesterov Y, Spokoiny V (2017) Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics* 17(2):527–566.

Nittala L, Erat S (2026) Designing knowledge-driven innovation contests. *Management Science* 72(2):1095–1111.

Park JS, Popowski L, Cai C, Morris MR, Liang P, Bernstein MS (2022) Social simulacra: Creating populated prototypes for social computing systems. *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, Article No.: 74.

Rajbhandari S, Li C, Yao Z, Zhang M, Aminabadi RY, Awan AA, Rasley J, He Y (2022) DeepSpeed-MoE: Advancing mixture-of-experts inference and training to power next-generation AI scale. *Proceedings of the 39th International Conference on Machine Learning*, 18332–18346.

Raschka S (2024) *Build A Large Language Model (From Scratch)* (Manning).

Simchi-Levi D, Mellou K, Menache I, Pathuri J (2026) Large language models for supply chain decisions. *AI in Supply Chains: Perspectives from Global Thought Leaders*, 93–104 (Springer).

Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D, et al. (2022) Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems 35*, 24824–24837.

Yuan J, Li H, Ding X, Xie W, Li YJ, Zhao W, Wan K, Shi J, Hu X, Liu Z (2025) Understanding and mitigating numerical sources of nondeterminism in LLM inference. URL https://arxiv.org/abs/2506.09501.

Zhang Y, Zhou Y, Ji K, Zavlanos MM (2022) A new one-point residual-feedback oracle for black-box learning and control. *Automatica* 136:110006.

Zhou C, Yang J, Xin L, Chen Y, He Z, Ge D (2025) Auto-formulating dynamic programming problems with large language models. https://arxiv.org/abs/2507.11737.

Supplemental Material

EC.1. Details of the Supply Chain Example

EC.1.1. System State

Similar to the retailer (R) discussed in Section 3.2, the elements of system state $\xi^{(t)}$ relevant to the manufacturer (M) and the consumer (C) are detailed as follows.

Manufacturer. The manufacturer's action $A_{\mathsf{M}}^{(t)}$ consists of a wholesale price $\texttt{WS}^{(t)}$ and an investment level in low-carbon technology $\texttt{TECH}^{(t)}$. These are chosen based only on information from the previous round, $H_{\mathsf{M}}^{(t-1)}$. After acting, the manufacturer observes the carbon emissions $\texttt{EMS}^{(t)}$ and communicates the wholesale price $\texttt{WS}^{(t)}$ and the carbon footprint $\texttt{FP}^{(t)}$ (the percentage reduction in emissions relative to a baseline) to the retailer. Later, after the other two agents have acted, the manufacturer receives the purchasing quantity $\texttt{QUT}^{(t)}$, which is part of the consumer's action, via the retailer.

The manufacturer has no pre-action observation or pre-action communications; i.e., $O_{\mathsf{M}}^{(t,\mathsf{preA})} = M_{\mathsf{M}}^{(t,\mathsf{preA})} = \varnothing$. Its post-action observation is $O_{\mathsf{M}}^{(t,\mathsf{postA})} = \texttt{EMS}^{(t)}$. Its post-action communications include the message sent to the retailer $M_{\mathsf{M}\to\mathsf{R}}^{(t,\mathsf{postA})} = (\texttt{WS}^{(t)}, \texttt{FP}^{(t)})$ and the message later received from the retailer $M_{\mathsf{M}\leftarrow\mathsf{R}}^{(t,\mathsf{postA})} = \texttt{QUT}^{(t)}$. Thus, at the end of this round, the manufacturer's memory module stores

$$H_{\mathsf{M}}^{(t)} = (\underbrace{\texttt{TECH}^{(t)}}_{A_{\mathsf{M}}^{(t)}}, \overbrace{\underbrace{\texttt{WS}^{(t)}}_{}, \texttt{FP}^{(t)}}^{M_{\mathsf{M}\to\mathsf{R}}^{(t,\mathsf{postA})}}, \underbrace{\texttt{EMS}^{(t)}}_{O_{\mathsf{M}}^{(t,\mathsf{postA})}}, \underbrace{\texttt{QUT}^{(t)}}_{M_{\mathsf{M}\leftarrow\mathsf{R}}^{(t,\mathsf{postA})}}).$$

Consumer. The consumer's action $A_{\mathsf{C}}^{(t)}$ consists of a purchasing quantity $\texttt{QUT}^{(t)}$ and the willingness to pay $\texttt{WTP}^{(t)}$. These are chosen using the information from the previous round, $H_{\mathsf{C}}^{(t-1)}$ and the pre-action message received from the retailer $M_{\mathsf{C}\leftarrow\mathsf{R}}^{(t,\mathsf{preA})}$. After acting, the consumer communicates the purchasing quantity $\texttt{QUT}^{(t)}$ to the retailer.

The consumer has no pre-action observation; i.e., $O_{\mathsf{C}}^{(t,\mathsf{preA})} = \varnothing$. Its pre-action communication is the message from the retailer $M_{\mathsf{C}\leftarrow\mathsf{R}}^{(t,\mathsf{preA})} = (\texttt{RT}^{(t)}, \texttt{AD}^{(t)})$. Its post-action communication is the message sent to the retailer $M_{\mathsf{C}\to\mathsf{R}}^{(t,\mathsf{postA})} = \texttt{QUT}^{(t)}$. There is no post-action observation. Thus, at the end of this round, the consumer's memory module stores

$$H_{\mathsf{C}}^{(t)} = (\underbrace{\texttt{AD}^{(t)}, \texttt{RT}^{(t)}}_{M_{\mathsf{C}\leftarrow\mathsf{R}}^{(t,\mathsf{preA})}}, \underbrace{\texttt{WTP}^{(t)}, \overbrace{\texttt{QUT}^{(t)}}^{M_{\mathsf{C}\to\mathsf{R}}^{(t,\mathsf{postA})}}}_{A_{\mathsf{C}}^{(t)}}).$$

EC.1.2. Objective Function

Following the notations in Example 1, the policymaker's objective function is given by

$$F(\theta;\xi) = -\big(\,\texttt{SCWF}(\theta;\xi) - \texttt{FISC}(\theta;\xi) - \texttt{ENV}(\xi)\big),$$

which captures trade-offs among supply chain welfare (SCWF), fiscal impact (FISC), and environmental externalities (ENV).

(i) The supply chain welfare is $\texttt{SCWF}(\theta;\xi) = \texttt{MPF} + \texttt{RPF} + \texttt{CS}$. Specifically, the manufacturer's profit is

$$\texttt{MPF} = (\texttt{WS} - c^{\text{prod}})\,\texttt{QUT} - 0.5c^{\text{tech}}\,\texttt{TECH}^2 - \theta_1\,\texttt{EMS}\cdot\texttt{QUT},$$

where c^{prod} and c^{tech} are the per unit cost of production and low-carbon technology adoption, respectively. The retailer's profit is

$$\texttt{RPF} = (\texttt{RT} - \texttt{WS})\,\texttt{QUT} - \texttt{MKT}\,.$$

The consumer surplus is

$$\texttt{CS} = (\texttt{WTP} - \texttt{RT} + \theta_2)\,\texttt{QUT}\,.$$

(ii) The fiscal impact is measured via an asymmetric function, given by

$$\texttt{FISC}(\theta;\xi) = ((\texttt{EXP} - c^{\text{tag}})^+)^{c_+} + ((c^{\text{tag}} - \texttt{EXP})^+)^{c_-}.$$

In particular, the policy expenditure is computed as

$$\texttt{EXP} = -\theta_1\,\texttt{EMS}\cdot\texttt{QUT} + \theta_2\,\texttt{QUT},$$

where the first term is the (negative) total emission taxes charged on the manufacturer, and the second term is the total subsidies granted to the consumer. The parameter c^{tag} denotes the target spending, and c_+ and c_- are the penalty intensities for overspending $(\texttt{EXP} - c^{\text{tag}})^+$ and underspending $(c^{\text{tag}} - \texttt{EXP})^+$, respectively; usually, $c_+ > c_-$ meaning overspending is penalized more heavily.

(iii) The environmental externality is computed as

$$\texttt{ENV}(\xi) = c_1^{\text{env}}(\texttt{EMS}\cdot\texttt{QUT})^{c_2^{\text{env}}},$$

where the parameter c_1^{env} represents the monetary cost of emissions, and c_2^{env} controls the curvature. The carbon emissions evolve according to $\texttt{EMS}^{(t+1)} = (\texttt{EMS}^{(t)} - \Delta\,\texttt{EMS}^{(t)})^+$, which remains nonnegative. In particular,

$$\Delta\,\texttt{EMS}^{(t)} = e^{\text{red}}(\texttt{EMS}^{(t)}(1+\zeta) - e^{\text{base}})\log\big(1 + \texttt{TECH}^{(t)}\big).$$

Here, t corresponds to the round index in LLM-MAS. The parameter e^{base} represents the minimal achievable emission level and e^{red} is the efficiency of emission reduction per unit of technological effort. The random variable $\zeta \sim N(0,\sigma^2)$ captures exogenous uncertainty in the emission process. The reduced emissions $\Delta\,\texttt{EMS}^{(t)}$ depend on the extent to which the realized emissions exceed e^{base}, which is scaled by a logarithmic term of technological effort $\texttt{TECH}^{(t)}$. This captures diminishing returns to technology: initial investments yield significant reductions, whereas subsequent investments provide progressively smaller gains.

EC.1.3. Agent Attributes

The attributes χ_i in the agent's profile module are configured to capture key behavioral characteristics. For the retailer, we include "willingness to collaborate", which is equally likely to be high, moderate, or low. This reflects the retailer's tendency to work with the manufacturer on promoting technology adoption through marketing efforts. For the consumer, we include "sustainability awareness", which is likewise equally likely to be eco-aware, eco-neutral, or eco-skeptical. This captures the degree to which the consumer is inclined to purchase sustainable products.

EC.1.4. Experimental Configurations

The parameters in $F(\theta;\xi)$ are specified as follows. The parameters related to $\texttt{SCWF}(\theta;\xi)$ are generated as $c^{\text{prod}} = 1 + \text{Unif}[0,1]$ and $c^{\text{tech}} \sim \text{Unif}[0.5,1]$; both are sampled once at initialization and then kept fixed throughout the experiment. The parameters related to $\texttt{FISC}(\theta;\xi)$ are set to $c_+ = 1.2, c_- = 0.8$, and $c^{\text{tag}} = 0$. The emission-related parameters are set to $\texttt{EMS}^{(0)} = e^{\text{init}} = 8$, $e^{\text{red}} = 0.05$, $e^{\text{base}} = 3$, and $\sigma = 0.5$. Finally, the parameters related to $\texttt{ENV}(\xi)$ are set to $c_1^{\text{env}} = 0.05$ and $c_2^{\text{env}} = 1.2$.

The agents' action spaces are defined as follows. For the manufacturer, $\texttt{TECH} \in [2,5]$ and $\texttt{WS} \in [6,8]$; for the retailer, $\texttt{MKT} \in [20,30]$ and $\texttt{RT} \in [12,15]$; and for the consumer, $\texttt{WTP} \in [15,18]$ and $\texttt{QUT} \in [5,15]$. Advertising expenses are categorized into three levels, taking the values of 15, 20, and 25 for low-, medium-, and high-quality advertisements, respectively.

The hyperparameters of the OTL algorithm are set to $\alpha = 0.75$, $\beta = 1$, and $\rho = 0.9$. BO is implemented using a Gaussian process with a Matérn kernel (smoothness 2.5, lengthscale 1) as the surrogate model and expected improvement as the acquisition function. The hyperparameters of LLaMA3.1-8B are configured as follows: "temperature" is set to 1, "top_p" is set to 0.9, and "max_new_tokens" is capped at 1000.

EC.2. Details of the Case Study

As discussed in Section 7.1, under the analytical model of Liu et al. (2018), the optimal design parameters admit closed-form expressions, given by (18). A key quantity in this design is the cutoff $t^*(K)$ that separates low- and high-ability contestants:

$$t^*(K) = \max\left\{ J^{-1}(t_0),\, F^{-1}\left(\left(\frac{NK}{V+NK}\right)^{\frac{1}{N-1}}\right)\right\},$$

where $J(t) = t - [1-F(t)]/f(t)$ is the virtual ability function that adjusts for information asymmetry resulting from the designer's imperfect knowledge of agent abilities, with $f > 0$ being the density corresponding to the cumulative distribution function F of abilities t. Here, J^{-1} and F^{-1} are the inverse functions of J and F, respectively, and $1/t_0$ is the marginal benefit of effort (money left in the budget) for the contest designer. For effort provision to be incentive-compatible, $1/t_0$ must exceed the minimum marginal cost of exerting effort $1/b$; otherwise, even the most capable contestant would abstain, and no contest should be held.

With the optimal design (18), the maximum expected total effort is

$$R^* = \int_{t^*(K)}^{b} \sum_{i=1}^{N} (J(t_i) - t_0)[(V + NK)F^{N-1}(t_i) - K]f(t_i)\,\mathrm{d}t_i + t_0 V. \tag{EC.2.1}$$

The model predicts that, under the optimal design, contestant i exerts effort e_i, depending on his ability t_i:

$$e_i = \begin{cases} 0, & t_i \le t^*(K), \\ t_i((V+NK)F^{N-1}(t_i) - K) - \displaystyle\int_{t^*(K)}^{t_i} \big((V+NK)F^{N-1}(s) - K\big)\,\mathrm{d}s, & t_i > t^*(K). \end{cases}$$

Thus, low-ability contestants exert zero effort, whereas high-ability contestants exert positive effort, which exceeds the reserve $\hat{e}^*$ and increases with t_i.

Huang et al. (2026) conducted lab experiments to evaluate six contest designs. These designs vary in the liability K and in the contest objective (total effort versus winner's effort), resulting in different entry fees, reserves, and shared prizes. Table EC.1 summarizes the specifications of these designs.

Table EC.1 Contest Designs in Huang et al. (2026)

Design	Liability K	Objective	Entry Fee E^*	Reserve $\hat{e}^*$	Shared Prize S^*
High_TW	40	Both	40	136.5	80
Medium_TW	10.5	Both	10.5	30.5	50.5
Low_W	2	Winner	2	35.5	10
Low_T	2	Total	2	4.5	42
Zero_W	0	Winner	0	36.5	0
Zero_T	0	Total	0	0	N.A.

Note. High, Medium, Low, and Zero indicate the level of K, and T and W indicate whether the optimization objective is total effort or winner's effort. Huang et al. (2026) prove that when K is large enough, the optimal designs for the two objectives are identical, which is case for "High_TW" and "Medium_TW". "Zero_T" is equivalent to a standard all-pay auction with no entry fee and no reserve; its optimal prize allocation is winner-takes-all, and prize sharing never occurs.

EC.3. Omitted Proofs

EC.3.1. Technical Lemmas

LEMMA EC.1. *Suppose the positive definite matrix $\Sigma(\xi)$ is given. Under Assumptions 1–2, for all $\theta \in \Theta$,*

$$\mathbb{E}\left[G^{\mathrm{GP}}(\theta; \xi, \xi^{\pm}, u, \delta)\right] = \Sigma(\xi)\nabla f_{\delta,\Lambda}(\theta), \quad u \sim \Lambda = N(0, \Sigma(\xi)).$$

Proof of Lemma EC.1. By definition, the smoothed gradient can be expanded as follows:

$$\begin{aligned} \nabla f_{\delta,\Lambda}(\theta) &= \frac{1}{\sqrt{(2\pi)^d |\Sigma(\xi)|}} \int \exp\left(-\frac{1}{2} u^{\intercal} \Sigma(\xi)^{-1} u\right) \nabla_\theta f(\theta + \delta u)\,\mathrm{d}u = \mathbb{E}_{u\sim\Lambda}[\nabla f(\theta + \delta u)] \\ &= \frac{\delta^{-d}}{\sqrt{(2\pi)^d |\Sigma(\xi)|}} \int \exp\left(-\frac{1}{2\delta^2} (\vartheta - \theta)^{\intercal} \Sigma(\xi)^{-1} (\vartheta - \theta)\right) \nabla_\vartheta f(\vartheta)\,\mathrm{d}\vartheta \\ &= \frac{\delta^{-d}}{\sqrt{(2\pi)^d |\Sigma(\xi)|}} \exp\left(-\frac{1}{2\delta^2} (\vartheta - \theta)^{\intercal} \Sigma(\xi)^{-1} (\vartheta - \theta)\right) f(\vartheta)\Big|_{-\infty}^{+\infty} \end{aligned}$$

$$
\begin{aligned}
&- \frac{\delta^{-d}}{\sqrt{(2\pi)^d|\Sigma(\xi)|}} \int f(\vartheta)\nabla_\vartheta \exp\left(-\frac{1}{2\delta^2}(\vartheta-\theta)^\intercal\Sigma(\xi)^{-1}(\vartheta-\theta)\right) \mathrm{d}\vartheta \\
&= \frac{\delta^{-(d+2)}}{\sqrt{(2\pi)^d|\Sigma(\xi)|}} \int f(\vartheta) \exp\left(-\frac{1}{2\delta^2}(\vartheta-\theta)^\intercal\Sigma(\xi)^{-1}(\vartheta-\theta)\right) \Sigma(\xi)^{-1}(\vartheta-\theta)\,\mathrm{d}\vartheta \\
&= \frac{\delta^{-2}}{\sqrt{(2\pi)^d|\Sigma(\xi)|}} \int f(\theta+\delta u) \exp\left(-\frac{1}{2}u^\intercal\Sigma(\xi)^{-1}u\right) \Sigma(\xi)^{-1}(\delta u)\,\mathrm{d}u \\
&= \Sigma(\xi)^{-1} \int \frac{1}{\sqrt{(2\pi)^d|\Sigma(\xi)|}} \exp\left(-\frac{1}{2}u^\intercal\Sigma(\xi)^{-1}u\right) \frac{f(\theta+\delta u)}{\delta} u\,\mathrm{d}u \\
&= \Sigma(\xi)^{-1}\,\mathbb{E}_{u\sim\Lambda}\left[\frac{f(\theta+\delta u)}{\delta}u\right] \\
&= \Sigma(\xi)^{-1}\,\mathbb{E}_{u\sim\Lambda}\left[\mathbb{E}_{\xi^+\sim\mu_{\theta+\delta u}}\left[\frac{F(\theta+\delta u;\xi^+)}{\delta}u\middle| u\right]\right] \\
&= \Sigma(\xi)^{-1}\,\mathbb{E}_{u\sim\Lambda}\left[\mathbb{E}_{\xi^\pm\sim\mu_{\theta\pm\delta u}}\left[\frac{F(\theta+\delta u;\xi^+)-F(\theta-\delta u;\xi^-)}{2\delta}u\middle| u\right]\right] \\
&= \Sigma(\xi)^{-1}\,\mathbb{E}[G^{\mathrm{GP}}(\theta;\xi,\xi^\pm,u,\delta)],
\end{aligned}
$$

where the interchange of integration and differentiation follows from the L-smoothness of f and the compactness of Θ. Here, $\exp\left(-\frac{1}{2\delta^2}(\vartheta-\theta)^\intercal\Sigma(\xi)^{-1}(\vartheta-\theta)\right) f(\vartheta)\big|_{-\infty}^{+\infty} = 0$ holds because the Gaussian tail dominates. The last equality is due to the symmetry of Gaussian. The bias of the GP gradient estimator is thus given by $\|\mathbb{E}\left[G^{\mathrm{GP}}(\theta;\xi,\xi^\pm,u,\delta)\right] - \nabla f_{\delta,\Lambda}(\theta)\| = \|(\Sigma(\xi)-I_d)\nabla f_{\delta,\Lambda}(\theta)\|$. $\square$

LEMMA EC.2. *Under Assumptions 4–7, there exists a constant $L_{\mathsf{P}\nu} > 0$ such that*

(i) $\sup_{\theta\in\Theta}\|\mathsf{P}_\theta\nu_\theta(\xi)\| \le L_{\mathsf{P}\nu}(1+\|\xi\|)$ *for all $\xi\in\Xi$; and*

(ii) $\|\mathsf{P}_\theta\nu_\theta(\xi) - \mathsf{P}_{\theta'}\nu_{\theta'}(\xi)\| \le L_{\mathsf{P}\nu}\|\theta-\theta'\|(1+\|\xi\|)$ *for all $\theta,\theta'\in\Theta$ and all $\xi\in\Xi$.*

Here, we write $\mathsf{P}_\theta f(\xi) = \int \mathsf{P}_\theta(\xi,\mathrm{d}\xi')f(\xi')$ for any measurable function f.

Proof of Lemma EC.2. Assumption 5 implies that the following drift condition holds: there exists a Lyapunov function $V:\Xi\mapsto[1,+\infty)$, independent of θ, such that $\mathsf{P}_\theta V(\xi) \le C_V V(\xi) + C_V'$ for some positive constants $C_V < 1$ and $C_V' < \infty$. See Meyn and Tweedie (2012, Chapter 16).

For (i), observe that $\|\mathsf{P}_\theta\nu_\theta(\xi)\| \le L_\nu \mathsf{P}_\theta(1+\|\xi\|) \le L_\nu(1+C_V\|\xi\|+C_V') \le L_{\mathsf{P}\nu}(1+\|\xi\|)$, where $L_{\mathsf{P}\nu} \coloneqq L_\nu\max\{C_V, 1+C_V'\}$. The first inequality is due to Assumption 7-(ii), and the second inequality is by the drift condition.

For (ii), by triangle inequality, $\|\mathsf{P}_\theta\nu_\theta(\xi) - \mathsf{P}_{\theta'}\nu_{\theta'}(\xi)\| \le \|\mathsf{P}_\theta\nu_\theta(\xi) - \mathsf{P}_{\theta'}\nu_\theta(\xi)\| + \|\mathsf{P}_{\theta'}\nu_\theta(\xi) - \mathsf{P}_{\theta'}\nu_{\theta'}(\xi)\|$. For the first part, we have

$$
\begin{aligned}
&\|\mathsf{P}_\theta\nu_\theta(\xi) - \mathsf{P}_{\theta'}\nu_\theta(\xi)\| = \left\|\int \nu_\theta(\xi')\left(\mathsf{P}_\theta\nu_\theta - \mathsf{P}_{\theta'}\nu_\theta\right)(\xi,\mathrm{d}\xi')\right\| \\
&\le \left(\sup_{\xi\neq\xi'}\frac{\|\nu_\theta(\xi)-\nu_\theta(\xi')\|}{\|\xi-\xi'\|}\right)\cdot\mathcal{W}_1(\mathsf{P}_\theta(\xi,\cdot),\mathsf{P}_{\theta'}(\xi,\cdot)) = \mathrm{Lip}(\nu_\theta)\cdot\mathcal{W}_1(\mathsf{P}_\theta(\xi,\cdot),\mathsf{P}_{\theta'}(\xi,\cdot)) \\
&\le L_\nu L_{\mathsf{P}}\|\theta-\theta'\|, \qquad \text{(EC.3.1)}
\end{aligned}
$$

where the first inequality follows from the Kantorovich-Rubinstein duality, and the second inequality is by Assumption 4 and Assumption 7-(i). For the second part, we have

$$\begin{aligned}\|\mathsf{P}_{\theta'}\nu_\theta(\xi)-\mathsf{P}_{\theta'}\nu_{\theta'}(\xi)\| &= \left\|\int(\nu_\theta(\xi')-\nu_{\theta'}(\xi'))\,\mathsf{P}_{\theta'}(\xi,\mathrm{d}\xi')\right\| \\ \le \mathsf{P}_{\theta'}\left(L_\nu\|\theta-\theta'\|(1+\|\xi\|)\right) &\le L_\nu\|\theta-\theta'\|(1+C_V\|\xi\|+C_V') \le L_{\mathsf{P}\nu}\|\theta-\theta'\|(1+\|\xi\|),\end{aligned}$$

where the first inequality is by Assumption 7-(iii) and the second inequality follows from (i). Combining the two parts completes the proof, where $L_{\mathsf{P}\nu} \coloneqq L_\nu L_{\mathsf{P}}$. $\square$

For the subsequent analysis, we define $\hat{f}(\theta;\xi) = \int_\Xi F(\theta;\xi')\mathsf{P}_\theta(\xi,\mathrm{d}\xi')$ and $\hat{f}_{\delta,\Lambda}(\theta;\xi) = \int_{\mathbb{R}^d}\hat{f}(\theta+\delta u;\xi)\,\Lambda(\mathrm{d}u)$. Here, $\hat{f}(\theta;\xi)$ is the expected value of $F(\theta;\xi')$ with ξ' following the transition distribution $\mathsf{P}_\theta(\xi,\cdot)$ given the current state ξ, and $\hat{f}_{\delta,\Lambda}(\theta;\xi)$ denotes its smoothed counterpart with $\Lambda = N(0,I_d)$. This construction parallels $f(\theta)$ and $f_\delta(\theta)$ in (1) and (4), except that the expectation is now taken over P_θ in lieu of the steady-state distribution μ_θ.

LEMMA EC.3. *Suppose Assumptions 3–7 hold, and let $\Lambda = N(0,I_d)$. Then for any $\xi\in\Xi$,*

$$\left\|\mathbb{E}_{u\sim\Lambda,\xi^\pm\sim\mathsf{P}_{\theta\pm\delta u(\xi,\cdot)}}[G(\theta;\xi^\pm,u,\delta)] - \mathbb{E}_{\xi'\sim\mathsf{P}_\theta(\xi,\cdot)}[H(\theta;\xi')]\right\| \le \delta L_{\hat{f}}(d+3)^{3/2}/2,$$

where $L_{\hat{f}}>0$ is some constant.

Proof of Lemma EC.3. We first show that $H(\theta;\xi)$ is Lipschitz continuous in θ for any given $\xi\in\Xi$. Recall the Poisson equation $H(\theta;\xi)-\nabla f(\theta) = \nu_\theta(\xi) - \mathbb{E}_{\xi'\sim\mathsf{P}_\theta(\xi,\cdot)}[\nu_\theta(\xi')]$. By triangle inequality, we have

$$\begin{aligned}\|H(\theta;\xi)-H(\theta';\xi)\| &= \left\|\left(\nu_\theta(\xi)-\mathbb{E}_{\xi'\sim\mathsf{P}_\theta(\xi,\cdot)}[\nu_\theta(\xi')]\right)-\left(\nu_{\theta'}(\xi)-\mathbb{E}_{\xi'\sim\mathsf{P}_{\theta'}(\xi,\cdot)}[\nu_{\theta'}(\xi')]\right)\right\| \\ &\le \underbrace{\|\nu_\theta(\xi)-\nu_{\theta'}(\xi)\|}_{\coloneqq(\mathrm{I})} + \underbrace{\left\|\mathbb{E}_{\xi'\sim\mathsf{P}_\theta(\xi,\cdot)}[\nu_\theta(\xi')]-\mathbb{E}_{\xi'\sim\mathsf{P}_{\theta'}(\xi,\cdot)}[\nu_{\theta'}(\xi')]\right\|}_{\coloneqq(\mathrm{II})}.\end{aligned}$$

Bound term (I). It is easy to show that $(\mathrm{I}) \le L_\nu(1+C_\xi)\|\theta-\theta'\|$, where $C_\xi = \|\xi\|$ by Assumption 7-(iii).

Bound term (II). Observe that

$$(\mathrm{II}) = \left\|\mathbb{E}_{\mathsf{P}_\theta}[\nu_\theta]-\mathbb{E}_{\mathsf{P}_{\theta'}}[\nu_{\theta'}]\right\| \le \underbrace{\left\|\mathbb{E}_{\mathsf{P}_\theta}[\nu_\theta-\nu_{\theta'}]\right\|}_{\coloneqq(\mathrm{III})} + \underbrace{\left\|\mathbb{E}_{\mathsf{P}_\theta}[\nu_{\theta'}]-\mathbb{E}_{\mathsf{P}_{\theta'}}[\nu_{\theta'}]\right\|}_{\coloneqq(\mathrm{IV})},$$

where P_θ and $\mathsf{P}_{\theta'}$ are shorthands for $\xi'\sim\mathsf{P}_\theta(\xi,\cdot)$ and $\xi'\sim\mathsf{P}_{\theta'}(\xi,\cdot)$, respectively. By Assumption 7-(iii), we have $(\mathrm{III}) \le L_\nu(1+C_\xi)\|\theta-\theta'\|$. Applying the Kantorovich-Rubinstein duality, we obtain $(\mathrm{IV}) \le \mathrm{Lip}(\nu_{\theta'})\cdot\mathcal{W}_1(\mathsf{P}_\theta(\xi,\cdot),\mathsf{P}_{\theta'}(\xi,\cdot)) \le L_\nu L_{\mathsf{P}}\|\theta-\theta'\|$ by Assumption 7-(i). Combining the terms (I), (III) and (IV) gives us $\|H(\theta;\xi)-H(\theta';\xi)\| \le L_H\|\theta-\theta'\|$, where $L_H = 2L_\nu(1+C_\xi)+L_\nu L_{\mathsf{P}}$. It then follows from Assumption 6 that

$$\left\|\nabla_\theta\hat{f}(\theta;\xi)-\nabla_{\theta'}\hat{f}(\theta';\xi)\right\| = \left\|\nabla_\theta\left(\int_\Xi F(\theta;\xi')\mathsf{P}_\theta(\xi,\mathrm{d}\xi')\right)-\nabla_{\theta'}\left(\int_\Xi F(\theta';\xi')\mathsf{P}_{\theta'}(\xi,\mathrm{d}\xi')\right)\right\|$$

$$
\begin{aligned}
&= \left\|\mathbb{E}_{\xi'\sim \mathsf{P}_\theta(\xi,\cdot)}[H(\theta;\xi')] - \mathbb{E}_{\xi'\sim \mathsf{P}_{\theta'}(\xi,\cdot)}[H(\theta';\xi')]\right\| \\
&\le \left\| \int \left(H(\theta;\xi') - H(\theta';\xi')\right) \mathsf{P}_\theta(\xi, \mathrm{d}\xi') \right\| + \left\| \int H(\theta';\xi')(\mathsf{P}_\theta - \mathsf{P}_{\theta'})(\xi, \mathrm{d}\xi') \right\| \\
&\le L_H \|\theta - \theta'\| + L_H \mathcal{W}_1\left(\mathsf{P}_\theta(\xi,\cdot), \mathsf{P}_{\theta'}(\xi,\cdot)\right) \\
&\le L_{\hat{f}} \|\theta - \theta'\|, \qquad \text{(EC.3.2)}
\end{aligned}
$$

where the penultimate inequality follows once again from the Kantorovich-Rubinstein duality, and in the last inequality we take $L_{\hat{f}} = L_H(1 + L_{\mathsf{P}})$. Note that

$$
\begin{aligned}
&\mathbb{E}_{u\sim\Lambda, \xi^\pm \sim \mathsf{P}_{\theta\pm\delta u(\xi,\cdot)}}[G(\theta;\xi^\pm,u,\delta)] \\
&= \int_{\mathbb{R}^d} \int_{\Xi\times\Xi} \frac{F(\theta+\delta u;\xi^+) - F(\theta-\delta u;\xi^-)}{2\delta} u \, \mathsf{P}_{\theta+\delta u}(\xi, \mathrm{d}\xi^+) \, \mathsf{P}_{\theta-\delta u}(\xi, \mathrm{d}\xi^-) \, \Lambda(\mathrm{d}u) \\
&= \int_{\mathbb{R}^d} \int_{\Xi} \frac{F(\theta+\delta u;\xi^+)}{\delta} u \, \mathsf{P}_{\theta+\delta u}(\xi, \mathrm{d}\xi^+) \, \Lambda(\mathrm{d}u) = \frac{1}{\delta} \int_{\mathbb{R}^d} \hat{f}(\theta+\delta u;\xi) u \, \Lambda(\mathrm{d}u) \qquad \text{(EC.3.3)} \\
&= -\frac{1}{\delta} \int_{\mathbb{R}^d} \hat{f}(\theta+\delta u;\xi) \nabla p_\Lambda(u) \, \mathrm{d}u = \frac{1}{\delta} \int_{\mathbb{R}^d} \nabla_u \hat{f}(\theta+\delta u;\xi) p_\Lambda(u) \, \mathrm{d}u \\
&= \frac{1}{\delta} \int_{\mathbb{R}^d} (\delta I_d)^\intercal \nabla_\theta \hat{f}(\theta+\delta u;\xi) \, p_\Lambda(u) \, \mathrm{d}u = \nabla_\theta \left(\int_{\mathbb{R}^d} \hat{f}(\theta+\delta u;\xi) \, \Lambda(\mathrm{d}u) \right) = \nabla_\theta \hat{f}_{\delta,\Lambda}(\theta;\xi),
\end{aligned}
$$

where the third line follows from the definition of $\hat{f}(\theta;\xi)$, and the fourth line is by Stein's identity with Gaussian tail. Hence,

$$
\begin{aligned}
&\left\|\mathbb{E}_{u\sim\Lambda, \xi^\pm \sim \mathsf{P}_{\theta\pm\delta u(\xi,\cdot)}}[G(\theta;\xi^\pm,u,\delta)] - \mathbb{E}_{\xi'\sim \mathsf{P}_\theta(\xi,\cdot)}[H(\theta;\xi')]\right\| = \left\|\nabla \hat{f}_{\delta,\Lambda}(\theta;\xi) - \nabla \hat{f}(\theta;\xi)\right\| \\
&= \left\| \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} \left(\frac{\hat{f}(\theta+\delta u;\xi) - \hat{f}(\theta;\xi)}{\delta} - \langle \nabla \hat{f}(\theta;\xi), u \rangle \right) u e^{-\frac{1}{2}\|u\|^2} \, \mathrm{d}u \right\| \\
&\le \frac{1}{(2\pi)^{d/2}\delta} \int_{\mathbb{R}^d} \left| \hat{f}(\theta+\delta u;\xi) - \hat{f}(\theta;\xi) - \langle \nabla \hat{f}(\theta;\xi), \delta u \rangle \right| \|u\| e^{-\frac{1}{2}\|u\|^2} \, \mathrm{d}u \\
&\le \frac{1}{(2\pi)^{d/2}\delta} \int_{\mathbb{R}^d} \frac{L_{\hat{f}}}{2} \|\delta u\|^2 \|u\| e^{-\frac{1}{2}\|u\|^2} \, \mathrm{d}u = \frac{\delta L_{\hat{f}}}{2} \mathbb{E}[\|u\|^3] \\
&\le \frac{\delta L_{\hat{f}}}{2} (d+3)^{3/2},
\end{aligned}
$$

where the second line uses (EC.3.3), $\mathbb{E}_{u\sim\Lambda}[u] = 0$ and $\mathbb{E}_{u\sim\Lambda}[uu^\intercal] = I_d$. The second inequality follows directly from the $L_{\hat{f}}$-smoothness of $\hat{f}(\theta;\xi)$ by (EC.3.2). The last inequality follows from an extension of Wendel's inequality to all $s > 1$, namely $\frac{\Gamma(x+s)}{\Gamma(x)} \le (x+s)^s$ for all $x > 0$. For integer $s = n$, this is immediate since $\frac{\Gamma(x+n)}{\Gamma(x)} = x(x+1)\cdots(x+n-1) \le (x+n)^n$. For general $s = n + r$ with $n = \lfloor s \rfloor$ and $0 \le r < 1$, $\frac{\Gamma(x+s)}{\Gamma(x)} = \frac{\Gamma(x+n+r)}{\Gamma(x+n)} \frac{\Gamma(x+n)}{\Gamma(x)} \le (x+n)^r (x+n)^n \le (x+s)^s$ by the log-convexity of Gamma function. Applying this inequality yields $\mathbb{E}[\|u\|^p] = 2^{p/2} \frac{\Gamma((d+p)/2)}{\Gamma(d/2)} \le 2^{p/2} \left(\frac{d+p}{2}\right)^{p/2} \le (d+p)^{p/2}$ for all $p \ge 3$. $\square$

EC.3.2. Proof of Asymptotic Convergence

Proof of Theorem 1 The proof builds upon Theorem 5.3.1 of Kushner and Clark (1978). Define the natural filtration associated with the OTL algorithm by $\mathcal{F}_k = \sigma(\theta_s, \xi_s : 0 \le s \le k; u_s, \xi_s^\pm : 0 \le s \le k-1)$ for

$k \geq 1$ with $\mathcal{F}_0 = \sigma(\theta_0, \xi_0)$. The sequence $\{\mathcal{F}_k\}_{k\geq 0}$ forms an increasing family of σ-algebras generated by all random variables observed up to iteration k.

Recall the update rule in the OTL algorithm (see Section 4.3):

$$\theta_{k+1} = \Pi_\Theta\left(\theta_k - \eta_k G(\theta_k; \xi_k^{\pm}, u_k, \delta_k)\right). \tag{EC.3.4}$$

It can be equivalently written as $\theta_{k+1} = \Pi_\Theta\left(\theta_k - \eta_k\left(\nabla f(\theta_k) + A_k + B_k + V_{k+1}\right)\right)$, where

$$A_k = \mathbb{E}[G(\theta_k; \xi_k^{\pm}, u_k, \delta_k)|\mathcal{F}_k] - H(\theta_k; \xi_{k+1}), \tag{EC.3.5}$$

$$B_k = H(\theta_k; \xi_{k+1}) - \nabla f(\theta_k), \tag{EC.3.6}$$

$$V_{k+1} = G(\theta_k; \xi_k^{\pm}, u_k, \delta_k) - \mathbb{E}[G(\theta_k; \xi_k^{\pm}, u_k, \delta_k)|\mathcal{F}_k]. \tag{EC.3.7}$$

Here, A_k denotes the approximation bias from the zeroth-order method; B_k denotes the Markovian bias associated with the decision-dependent transition kernel; and V_{k+1} denotes the martingale difference noise.

Define a sequence of time points t_k by $t_0 = 0$ and $t_k = \sum_{s=0}^{k-1}\eta_s$ for $k \geq 1$. For any (continuous) time $t \geq 0$, let $m(t)$ denote the iterate index identifying the discrete time interval $[t_k, t_{k+1})$; that is, $m(t) = \max\{k \geq 0 : t_k \leq t\}$. Our goal is to show that the terms A_k, B_k and V_{k+1} are asymptotically negligible, allowing the recursion to be analyzed through the limiting ordinary differential equation (ODE).

Approximation bias. From (EC.3.5), we decompose A_k into two terms:

$$A_k = \underbrace{\mathbb{E}\left[\frac{F(\theta_k + \delta_k u_k; \xi_k^+) - F(\theta_k - \delta_k u_k; \xi_k^-)}{2\delta_k} u_k \middle| \mathcal{F}_k\right] - \mathbb{E}_{\xi_{k+1}\sim \mathsf{P}_{\theta_k}(\xi_k,\cdot)}[H(\theta_k; \xi_{k+1})|\mathcal{F}_k]}_{:=A_k^{(\mathrm{I})}}$$
$$+ \underbrace{\mathbb{E}_{\xi_{k+1}\sim \mathsf{P}_{\theta_k}(\xi_k,\cdot)}[H(\theta_k; \xi_{k+1})|\mathcal{F}_k] - H(\theta_k; \xi_{k+1})}_{:=A_k^{(\mathrm{II})}}.$$

(1) Control term $A_k^{(\mathrm{I})}$. By Lemma EC.3 and Assumption 8, we have

$$\left\|\sum_{k=0}^{\infty}\eta_k A_k^{(\mathrm{I})}\right\| \leq \sum_{k=0}^{\infty}\eta_k\left\|\nabla \hat{f}_{\delta_k,\Lambda}(\theta_k;\xi_k) - \nabla \hat{f}(\theta_k;\xi_k)\right\| \leq \frac{L_{\hat{f}}}{2}(d+3)^{3/2}\sum_{k=0}^{\infty}\eta_k\delta_k < \infty,$$

where the bound in Lemma EC.3 holds for all $\xi_k \in \Xi$. Since the series converges absolutely, we then have $\eta_k A_k^{(\mathrm{I})} \to 0$ as $k \to \infty$.

(2) Control term $A_k^{(\mathrm{II})}$. Define $D_k = \sum_{s=0}^{k-1}\eta_s A_s^{(\mathrm{II})}$ for $k \geq 1$, and $D_0 = 0$. Since $\mathbb{E}[A_k^{(\mathrm{II})}|\mathcal{F}_k] = 0$, it follows that $\mathbb{E}[D_{k+1}|\mathcal{F}_k] = \mathbb{E}[D_k + \eta_k A_k^{(\mathrm{II})}|\mathcal{F}_k] = D_k$, and thus $(D_k, \mathcal{F}_k)$ is a martingale. Its quadratic variation satisfies

$$\sum_{k=0}^{\infty}\mathbb{E}[\|D_{k+1} - D_k\|^2|\mathcal{F}_k] = \sum_{k=0}^{\infty}\eta_k^2\,\mathbb{E}[\|A_k^{(\mathrm{II})}\|^2|\mathcal{F}_k]$$
$$= \sum_{k=0}^{\infty}\eta_k^2\,\mathbb{E}\left[\left\|\mathbb{E}_{\xi_{k+1}\sim \mathsf{P}_{\theta_k}(\xi_k,\cdot)}[H(\theta_k;\xi_{k+1})|\mathcal{F}_k] - H(\theta_k;\xi_{k+1})\right\|^2\middle|\mathcal{F}_k\right]$$

$$\leq 4\sum_{k=0}^{\infty}\eta_k^2\,\mathbb{E}_{\xi_{k+1}\sim \mathsf{P}_{\theta_k}(\xi_k,\cdot)}\left[\left\|H(\theta_k;\xi_{k+1})\right\|^2\right] \leq 4C_H\sum_{k=0}^{\infty}\eta_k^2 < \infty, \tag{EC.3.8}$$

where the last inequality follows from Assumption 7-(iv), together with the fact that η_k decays to zero faster than δ_k under Assumption 8. By martingale convergence theorem, D_k converges almost surely and in L^2. By Doob's L^2 maximal inequality, there exists a constant $\widetilde{C}_0 > 0$ such that for any integer $j \geq 0$ and any finite $T > 0$,

$$\mathbb{E}\left[\sup_{m(jT)\leq k\leq m((j+1)T)}\left\|D_k - D_{m(jT)}\right\|^2\right] \leq \widetilde{C}_0\,\mathbb{E}\left[\left\|D_{m((j+1)T)} - D_{m(jT)}\right\|^2\right] \to 0,$$

as $j \to \infty$. By Jensen's inequality,

$$\mathbb{E}\left[\sup_{m(jT)\leq k\leq m((j+1)T)}\left\|D_k - D_{m(jT)}\right\|\right] \leq \left(\mathbb{E}\left[\sup_{m(jT)\leq k\leq m((j+1)T)}\left\|D_k - D_{m(jT)}\right\|^2\right]\right)^{1/2} \to 0,$$

as $j \to \infty$. Given an $\epsilon > 0$, define the event $E_j(\epsilon) \coloneqq \left\{\sup_{m(jT)\leq k\leq m((j+1)T)}\left\|D_k - D_{m(jT)}\right\| \geq \epsilon\right\}$ for any integer $j \geq 0$. By Markov's inequality, for any $\epsilon > 0$,

$$\mathbb{P}\left(E_j(\epsilon)\right) \leq \epsilon^{-1}\,\mathbb{E}\left[\sup_{m(jT)\leq k\leq m((j+1)T)}\left\|D_k - D_{m(jT)}\right\|\right],$$

which is summable following similar arguments as in (EC.3.8). Then by the Borel-Cantelli lemma, $\mathbb{P}\left(E_j(\epsilon)\text{ i.o.}\right) = 0$, so almost surely, $\sup_{m(jT)\leq k\leq m((j+1)T)}\left\|D_k - D_{m(jT)}\right\| \to 0$ as $j \to \infty$. Equivalently,

$$\lim_{k\to\infty}\left(\sup_{j\geq k}\max_{t\leq T}\left\|\sum_{s=m(jT)}^{m(jT+t)-1}\eta_s A_s^{(\mathrm{II})}\right\|\right) = 0, \quad \text{a. s.}$$

Hence, the approximation bias is asymptotically negligible almost surely.

Markovian bias. From (11) and (EC.3.6), we decompose B_k into three terms:

$$\begin{aligned} B_k &= H(\theta_k;\xi_{k+1}) - \nabla f(\theta_k) = \nu_{\theta_k}(\xi_{k+1}) - \mathsf{P}_{\theta_k}\nu_{\theta_k}(\xi_{k+1}) \\ &= \underbrace{\left(\nu_{\theta_k}(\xi_{k+1}) - \mathsf{P}_{\theta_k}\nu_{\theta_k}(\xi_k)\right)}_{\coloneqq B_k^{(\mathrm{I})}} + \underbrace{\left(\mathsf{P}_{\theta_k}\nu_{\theta_k}(\xi_k) - \mathsf{P}_{\theta_{k+1}}\nu_{\theta_{k+1}}(\xi_{k+1})\right)}_{\coloneqq B_k^{(\mathrm{II})}} + \underbrace{\left(\mathsf{P}_{\theta_{k+1}}\nu_{\theta_{k+1}}(\xi_{k+1}) - \mathsf{P}_{\theta_k}\nu_{\theta_k}(\xi_{k+1})\right)}_{\coloneqq B_k^{(\mathrm{III})}}, \end{aligned}$$

where $B_k^{(\mathrm{I})}, B_k^{(\mathrm{II})}$, and $B_k^{(\mathrm{III})}$ represent martingale, coboundary, and residual term. Our goal is to show that there exists a constant $T > 0$ such that

$$\lim_{k\to\infty}\left(\sup_{j\geq k}\max_{t\leq T}\left\|\sum_{s=m(jT)}^{m(jT+t)-1}\eta_{s+1}B_s^{(\mathrm{i})}\right\|\right) = 0, \quad \text{a. s.} \quad \forall(\mathrm{i}) \in \{(\mathrm{I}),(\mathrm{II}),(\mathrm{III})\}. \tag{EC.3.9}$$

For convenience, define $S_k^{(\mathrm{i})} = \sum_{s=0}^{k-1}\eta_{s+1}B_s^{(\mathrm{i})}$ for $(\mathrm{i}) \in \{(\mathrm{I}),(\mathrm{II}),(\mathrm{III})\}$.

(1) Analyze martingale term $B_k^{(\mathrm{I})}$. It suffices to show that for some $T>0$,

$$\lim_{j\to\infty}\left(\sup_{m(jT)\le k\le m((j+1)T)}\left\|S_k^{(\mathrm{I})}-S_{m(jT)}^{(\mathrm{I})}\right\|\right)=0,\quad \text{a.s.}$$

Note that $\mathbb{E}[B_k^{(\mathrm{I})}|\mathcal{F}_k]=\mathbb{E}[\nu_{\theta_k}(\xi_{k+1})|\mathcal{F}_k]-\mathsf{P}_{\theta_k}\nu_{\theta_k}(\xi_k)=0$, so $S_k^{(\mathrm{I})}$ is a martingale adapted to $\mathcal{F}_k$. By Doob's L^2 maximal inequality, there exists a constant $\widetilde{C}_1>0$ such that

$$\begin{aligned}
&\mathbb{E}\left[\sup_{m(jT)\le k\le m((j+1)T)}\left\|S_k^{(\mathrm{I})}-S_{m(jT)}^{(\mathrm{I})}\right\|^2\right]\le\widetilde{C}_1\,\mathbb{E}\left[\left\|S_{m((j+1)T)}^{(\mathrm{I})}-S_{m(jT)}^{(\mathrm{I})}\right\|^2\right]\\
&\le\widetilde{C}_1\,\mathbb{E}\left[\sum_{s=m(jT)}^{m((j+1)T)-1}\eta_{s+1}^2\|B_s^{(\mathrm{I})}\|^2\right]=\widetilde{C}_1\sum_{s=m(jT)}^{m((j+1)T)-1}\eta_{s+1}^2\,\mathbb{E}[\|\nu_{\theta_s}(\xi_{s+1})-\mathsf{P}_{\theta_s}\nu_{\theta_s}(\xi_s)\|^2]\\
&\le 4\widetilde{C}_1\sum_{s=m(jT)}^{m((j+1)T)-1}\eta_{s+1}^2\,\mathbb{E}[\|\nu_{\theta_s}(\xi_{s+1})\|^2]\\
&\le 4\widetilde{C}_1L_\nu\sum_{s=m(jT)}^{m((j+1)T)-1}\eta_{s+1}^2\,\mathbb{E}[(1+\|\xi_{s+1}\|)^2]\coloneqq\widetilde{C}_2\sum_{s=m(jT)}^{m((j+1)T)-1}\eta_{s+1}^2\,\mathbb{E}\left[\|\xi_{s+1}\|^2\right]\\
&\le\widetilde{C}_2\widetilde{C}_3\sum_{s=m(jT)}^{m((j+1)T)-1}\eta_{s+1}^2\to 0,
\end{aligned}$$

as $j\to\infty$ since $\sum_{k=0}^{\infty}\eta_k^2<\infty$, where the last inequality follows from the drift condition $\mathbb{E}[\|\xi_{k+1}\|^2|\mathcal{F}_k]\le C_V\|\xi_k\|^2+C_V'$. Taking the total expectation and unrolling the recurrence, we have

$$\mathbb{E}[\|\xi_{k+1}\|^2]\le C_V^{k+1}\,\mathbb{E}[\|\xi_0\|^2]+C_V'\sum_{s=0}^{k}C_V^s\le C_V\,\mathbb{E}[\|\xi_0\|^2]+\frac{C_V'}{1-C_V}\coloneqq\widetilde{C}_3,\tag{EC.3.10}$$

which holds for all $k\ge 0$ since $0<C_V<1$. Again, by Jensen's inequality,

$$\mathbb{E}\left[\sup_{m(jT)\le k\le m((j+1)T)}\left\|S_k^{(\mathrm{I})}-S_{m(jT)}^{(\mathrm{I})}\right\|\right]\le\left(\mathbb{E}\left[\sup_{m(jT)\le k\le m((j+1)T)}\left\|S_k^{(\mathrm{I})}-S_{m(jT)}^{(\mathrm{I})}\right\|^2\right]\right)^{1/2}\to 0,$$

as $j\to\infty$. The remaining steps proceed by repeating the same reasoning as in the analysis for $\eta_k A_k^{(\mathrm{II})}$, and is therefore omitted for brevity.

(2) Analyze coboundary term $B_k^{(\mathrm{II})}$. We exploit the telescoping structure:

$$\begin{aligned}
S_k^{(\mathrm{II})}&=\sum_{s=0}^{k-1}\eta_{s+1}B_s^{(\mathrm{II})}=\sum_{s=0}^{k-1}\eta_{s+1}\left(\mathsf{P}_{\theta_s}\nu_{\theta_s}(\xi_s)-\mathsf{P}_{\theta_{s+1}}\nu_{\theta_{s+1}}(\xi_{s+1})\right)\\
&=\eta_1\mathsf{P}_{\theta_0}\nu_{\theta_0}(\xi_0)-\left[\sum_{s=1}^{k-1}(\eta_s-\eta_{s+1})\mathsf{P}_{\theta_s}\nu_{\theta_s}(\xi_s)\right]-\eta_k\mathsf{P}_{\theta_k}\nu_{\theta_k}(\xi_k).
\end{aligned}$$

For any $n\ge k$, it follows that

$$S_n^{(\mathrm{II})}-S_k^{(\mathrm{II})}=-\left[\sum_{s=k}^{n-1}(\eta_s-\eta_{s+1})\mathsf{P}_{\theta_s}\nu_{\theta_s}(\xi_s)\right]-\eta_n\mathsf{P}_{\theta_n}\nu_{\theta_n}(\xi_n)+\eta_k\mathsf{P}_{\theta_k}\nu_{\theta_k}(\xi_k).$$

By Lemma EC.2-(i), we have

$$\|S_n^{(\mathrm{II})} - S_k^{(\mathrm{II})}\| \le \underbrace{L_{\mathsf{P}\nu}\left[\sum_{s=k}^{n-1}(\eta_s - \eta_{s+1})(1+\|\xi_s\|)\right]}_{:=(\mathrm{I})} + \underbrace{\|\eta_n \mathsf{P}_{\theta_n}\nu_{\theta_n}(\xi_n)\| + \left\|\eta_k \mathsf{P}_{\theta_k}\nu_{\theta_k}(\xi_k)\right\|}_{:=(\mathrm{II})}.$$

The term (II) is asymptotically negligible almost surely since

$$\mathbb{E}\left[\sum_{s=0}^{\infty}\|\eta_s \mathsf{P}_{\theta_s}\nu_{\theta_s}(\xi_s)\|^2\right] \le L_{\mathsf{P}\nu}\,\mathbb{E}\left[\sum_{s=0}^{\infty}\eta_s^2(1+\|\xi_s\|)^2\right] \le L_{\mathsf{P}\nu}(1+\widetilde{C}_3)^2\sum_{s=0}^{\infty}\eta_s^2 < \infty,$$

by the bound in (EC.3.10). Then, by Tonelli's theorem, we have $\sum_{s=0}^{\infty}\mathbb{E}[\|\eta_s \mathsf{P}_{\theta_s}\nu_{\theta_s}(\xi_s)\|^2] < \infty$ and thus $\|\eta_s \mathsf{P}_{\theta_s}\nu_{\theta_s}(\xi_s)\| \to 0$ almost surely as $s \to \infty$.

For the term (I), we have

$$\mathbb{E}\left[\sum_{s=1}^{\infty}(\eta_s - \eta_{s+1})(1+\|\xi_s\|)\right] \le (1+\widetilde{C}_3)\sum_{s=1}^{\infty}(\eta_s - \eta_{s+1}) \le \widetilde{C}_4\eta_1 < \infty.$$

where the first inequality follows from (EC.3.10). Since $\eta_s > \eta_{s+1}$, Tonelli's theorem applies. Therefore,

$$\sup_{n\ge k}\left\|\sum_{s=k}^{n-1}\eta_{s+1}B_s^{(\mathrm{II})}\right\| = \sup_{n\ge k}\|S_n^{(\mathrm{II})} - S_k^{(\mathrm{II})}\| \lesssim \sup_{n\ge k}\sum_{s=k}^{n-1}(\eta_s - \eta_{s+1})(1+\|\xi_s\|) \to 0,$$

almost surely as $k \to \infty$.

(3) Analyze residual term $B_k^{(\mathrm{III})}$. By Lemma EC.2-(ii),

$$\begin{aligned}
\|B_{k+1}^{(\mathrm{III})}\| &= \left\|\mathsf{P}_{\theta_{k+1}}\nu_{\theta_{k+1}}(\xi_{k+1}) - \mathsf{P}_{\theta_k}\nu_{\theta_k}(\xi_{k+1})\right\| \le L_{\mathsf{P}\nu}\|\theta_{k+1} - \theta_k\|(1+\|\xi_{k+1}\|) \\
&= L_{\mathsf{P}\nu}\|\Pi_\Theta\left(\theta_k - \eta_k G(\theta_k;\xi_k^{\pm},u_k,\delta_k)\right) - \Pi_\Theta(\theta_k)\|(1+\|\xi_{k+1}\|) \\
&\le L_{\mathsf{P}\nu}\left\|\left(\theta_k - \eta_k\frac{F(\theta_k+\delta_k u_k;\xi_k^+) - F(\theta_k-\delta_k u_k;\xi_k^-)}{2\delta_k}u_k\right) - \theta_k\right\|(1+\|\xi_{k+1}\|) \\
&\le L_{\mathsf{P}\nu}C_F\left(\frac{\eta_k}{\delta_k}\right)\|u_k\|(1+\|\xi_{k+1}\|),
\end{aligned}$$

where the second inequality is by the nonexpansiveness of projection Π_Θ, and the last inequality is by Assumption 3. By Markov's inequality, for any $\epsilon > 0$,

$$\begin{aligned}
\mathbb{P}(\|B_{k+1}^{(\mathrm{III})}\| > \epsilon) &\le \left(\frac{L_{\mathsf{P}\nu}C_F}{\epsilon}\right)^2\left(\frac{\eta_k}{\delta_k}\right)^2\mathbb{E}[\|u_k\|^2(1+\|\xi_{k+1}\|)^2] \\
&\le \left(\frac{L_{\mathsf{P}\nu}C_F}{\epsilon}\right)^2 2d(1+\widetilde{C}_3)\left(\frac{\eta_k}{\delta_k}\right)^2 \to 0,
\end{aligned}$$

where the last inequality is due to (EC.3.10), and $\eta_k/\delta_k \to 0$ as $k \to \infty$. Here, $\mathbb{P}(\|B_{k+1}^{(\mathrm{III})}\| > \epsilon)$ is summable since $\sum_{k=0}^{\infty}\eta_k^2/\delta_k^2 < \infty$. Then by the Borel-Cantelli lemma, we have $\|B_{k+1}^{(\mathrm{III})}\| \to 0$ as $k \to \infty$ almost surely. Since $\eta_k \to 0$ as $k \to \infty$, the weighted tail sum is also asymptotically negligible; that is, as $k \to \infty$,

$$\sup_{n\ge k}\left\|\sum_{s=k}^{n-1}\eta_s B_s^{(\mathrm{III})}\right\| \to 0 \quad \text{a.s.}$$

Hence, the Markovian bias is asymptotically negligible almost surely.

Martingale difference sequence. From (EC.3.7), V_{k+1} is $\mathcal{F}_{k+1}$-measurable and satisfies $\mathbb{E}[V_{k+1}|\mathcal{F}_k]=0$ almost surely. Define $M_k=\sum_{s=0}^{k-1}\eta_s V_{s+1}$ for $k\geq 1$, and $M_0=0$. It follows that $M_{k+1}-M_k$ is a martingale difference sequence. Its quadratic variation is given by

$$\begin{aligned}
&\sum_{k=0}^{\infty}\mathbb{E}[\|M_{k+1}-M_k\|^2|\mathcal{F}_k]=\sum_{k=0}^{\infty}\eta_k^2\,\mathbb{E}[\|V_{k+1}\|^2|\mathcal{F}_k]\\
&=\sum_{k=0}^{\infty}\eta_k^2\,\mathbb{E}[\|G(\theta_k;\xi_k^{\pm},u_k,\delta_k)-\mathbb{E}[G(\theta_k;\xi_k^{\pm},u_k,\delta_k)|\mathcal{F}_k]\|^2|\mathcal{F}_k]\leq 4\sum_{k=0}^{\infty}\eta_k^2\,\mathbb{E}[\|G(\theta_k;\xi_k^{\pm},u_k,\delta_k)\|^2|\mathcal{F}_k]\\
&=\sum_{k=0}^{\infty}\frac{\eta_k^2}{\delta_k^2}\,\mathbb{E}\left[\left(F(\theta_k+\delta_k u_k;\xi_k^{+})-F(\theta_k-\delta_k u_k;\xi_k^{-})\right)^2\|u_k\|^2\Big|\,\mathcal{F}_k\right]\\
&\leq\sum_{k=0}^{\infty}\frac{\eta_k^2}{\delta_k^2}\,\mathbb{E}\left[(2C_F)^2\|u_k\|^2\big|\,\mathcal{F}_k\right]\leq 4dC_F^2\sum_{k=0}^{\infty}\frac{\eta_k^2}{\delta_k^2}<\infty,
\end{aligned}$$

where the last inequality is by Assumption 3. Consequently, M_k converges almost surely and in L^2. Then by Doob's L^2 maximal inequality, $\sup_{m(jT)\leq k\leq m((j+1)T)}\|M_k-M_{m(jT)}\|\to 0$ as $j\to\infty$ almost surely for any finite $T>0$. Using the arguments similar to those for $\eta_k A_k^{(\mathrm{II})}$, we have

$$\lim_{k\to\infty}\left(\sup_{j\geq k}\max_{t\leq T}\left\|\sum_{s=m(jT)}^{m(jT+t)-1}\eta_s V_{s+1}\right\|\right)=0,\quad \text{a. s.}$$

Hence, the martingale difference sequence is asymptotically negligible almost surely.

Limiting ODE. The continuous-time interpolation of the sequence $\{\theta_k\}_{k\geq 0}$ is defined by $\theta(t_k)=\theta_k$ for $t\in[t_k,t_{k+1})$ with $k\geq 0$. This construction yields a continuous-time interpolated trajectory that approximates the discrete-time dynamics of the OTL algorithm. In the limit as the stepsize $\eta_k\to 0$, the evolution of this interpolated trajectory is described by the ODE:

$$\dot{\theta}(t)=\overline{\Pi}_{\Theta}\left(-\nabla f(\theta(t))\right), \tag{EC.3.11}$$

where $\overline{\Pi}_{\Theta}(\cdot)$ denotes the directional derivative of the projection operator $\Pi_{\Theta}(\cdot)$. For any continuous function $\psi:\Theta\mapsto\mathbb{R}^d$ and any point $\vartheta\in\Theta$, it is given by $\overline{\Pi}_{\Theta}(\psi(\vartheta))=\lim_{\epsilon\downarrow 0}\left((\Pi(\vartheta+\epsilon\psi(\vartheta))-\vartheta)/\epsilon\right)$. Since Θ is compact and convex, the limit always exists. In particular, if ϑ lies in the interior of Θ, it follows trivially that $\overline{\Pi}(\psi(\vartheta))=\psi(\vartheta)$. If ϑ is located on the smooth boundary of Θ, the projection behaves locally as the orthogonal projection of $\psi(\vartheta)$ onto the tangent cone $\mathcal{T}_{\Theta}(\vartheta)$; i.e., $\overline{\Pi}(\psi(\vartheta))=\Pi_{\mathcal{T}_{\Theta}(\vartheta)}(\psi(\vartheta))$. In both cases, the limit is unique. $\overline{\Pi}(\cdot)$ may be set-valued when ϑ is not regular; i.e., a nonsmooth point of the boundary.

For any trajectory of the projected ODE in (EC.3.11) and any regular point $\theta(t)$, we have

$$\frac{d}{dt}f(\theta(t))=\nabla f(\theta)(t)^{\intercal}\dot{\theta}(t)=-\|\Pi_{\mathcal{T}_{\Theta}(\theta)}(\nabla f(\theta(t)))\|^2\leq 0.$$

Hence, $f(\theta(t))$ is non-increasing and thus a Lyapunov function for this projected gradient flow. Under Assumption 1, all trajectories are automatically bounded and remain within Θ. By LaSalle's invariance principle, every bounded trajectory converges to the largest invariant subset of $\Theta_0 \coloneqq \left\{\theta \in \Theta : \Pi_{\mathcal{T}_\Theta(\theta)}(\nabla f(\theta)) = 0\right\} = \{\theta \in \Theta : 0 \in \nabla f(\theta) + \mathcal{N}_\Theta(\theta)\}$, which is the set of all locally asymptotically stable points. For any given $\theta^* \in \Theta_0$, its attraction domain $D(\theta^*) = \{\theta_0 \in \Theta : \lim_{t\to\infty} \theta(t;\theta_0) = \theta^*\}$ is open. Therefore, there exists a small ball $\mathbb{B}(\theta^*, r) \subset D(\theta^*)$ containing infinitely many iterates θ_k.

Under Assumption 8 where $\sum_{k=0}^{\infty} \eta_k = \infty$ and $\sum_{k=0}^{\infty} \eta_k^2 < \infty$, we conclude that $\lim_{k\to\infty} \theta_k = \theta^*$. This establishes that when the trajectory converges to θ^*, and the stochastic recursion (EC.3.4) asymptotically tracks this ODE, then the sequence of the OTL algorithm iterates $\{\theta_k\}_{k\geq 0}$ also converges to θ^*. $\square$

References

Kushner HJ, Clark DS (1978) *Stochastic Approximation Methods for Constrained and Unconstrained Systems* (Springer).