CSE 6730 Final Project

LLM-Empowered Kudzu Modeling and Simulation

Team 20

Aaron Liu
Linyuan Yu
Tristan Thakur
Michael Wong
Jae Young Beck

GitHub repository:
https://github.gatech.edu/jbeck73/CSE6730Project

PROJECT FINAL REPORT: LLM-Empowered Kudzu Modeling and Simulation

Aaron Liu, Linyuan Yu, Tristan Thakur, Michael Wong, Jae Young Beck

SM1. Abstract. We develop an LLM-empowered simulator for managing the spread of kudzu, a fast-growing invasive vine in the southeastern United States. Our model couples a spatial Cellular Automata (CA) representation of kudzu dynamics with a Large Language Model (LLM) that plays the role of a budget-constrained manager. The CA tracks four life-cycle stages on a land-cover grid derived from NLCD data and parametrized following Aurambout and Endress. Each week, the CA updates local growth and spread; at the start of each month, the manager receives a compact field observation of the landscape and remaining budget and chooses spatially targeted treatment actions. We implemented three baselines (no intervention, a greedy hotspot rule, and a simple heuristic policy) and compared them against the LLM policy over a three-year (156-week) horizon in Clark County, Washington. In this scenario, the LLM eradicates kudzu (zero final abundance) while spending a moderate fraction of the available budget, whereas baseline policies either underspend and allow widespread infestation or overspend and still fail to eradicate.

SM2. Project Description. The overarching goal of this project is to build a simulation testbed in which an AI decision maker manages an ecological invasion under realistic constraints. We focus on kudzu (Pueraria montana), an invasive vine that spreads rapidly and can smother trees, infrastructure, and native vegetation. Kudzu is interesting from a modeling perspective because it has a clear life-cycle structure, spreads through both local vegetative growth and long-distance dispersal, and is notoriously expensive to control once established. Land managers must decide where and when to intervene, often with limited budgets and incomplete information.

Our simulation is designed to capture several features of this real-world phenomenon that we consider most relevant. First, the spread of kudzu is inherently spatial. Patches occupy some parcels of land but not others, and both spread and management actions are local. To represent this, we model the landscape as a two-dimensional grid derived from Clark County land-cover data, where each cell represents an area on the ground and carries a land-cover label such as forest, developed land, or water. The land-cover type determines how suitable the cell is for kudzu, which we encode as a carrying capacity. Second, kudzu passes through distinct life-cycle stages that differ in growth, mortality, and management response. We therefore track seeds in a seed bank, early seedlings, saplings, and adults. Third, spread is driven by both local expansion of vines into neighboring cells and less common long-distance dispersal events that create satellite infestations away from the origin patch. Finally, managers operate under budget constraints: each treatment costs money and they must balance current suppression against future risk.

The central question we ask is how a large language model behaves when placed in the role of such a manager. At a high level, our simulator runs forward in weekly steps. The CA updates the spatial kudzu every week using programmed ecological rules. Once per month, the simulation pauses and constructs a simplified field report summarizing what is happening on the landscape: total kudzu, the locations and intensities of the worst hotspots, and the remaining budget. This observation is sent to a policy, which can be either a hand-coded heuristic or the LLM. The policy chooses a set of actions such as treating a list of cells at a given intensity, removing seeds from particularly risky locations, or surveying a specific region of land to identify potential Kudzu establishment. These actions are then applied to the CA state subject to a fixed budget. In this way, we can compare an LLM-driven manager to simpler rule-based strategies in a controlled, transparent environment.

A video summary of the following project is also available here.

SM3. Literature Review.

SM3.1. Introduction and Background. Agent-based modeling and simulation (ABMS) has long been used to study how complex collective phenomena arise from the interactions of autonomous entities. Traditional ABMS systems rely on manually designed behavioral rules or statistical models, which restrict adaptability and realism. The recent emergence of large language models (LLMs) introduces a transformative opportunity: these models can act as cognitive cores for agents, capable of reasoning, communication, and contextual decision-making. The two papers reviewed here—Gao et al. (2024) [SM2] and Wu et al. (2025) [SM6] —together provide both a comprehensive overview and a critical assessment of this new research direction. We will utilize an LLM-based approach for a specific Kudzu application discussed later.

SM3.2. Existing Models and Techniques in LLM-Based ABMS. Gao et al. (2024) [SM2] provide the first comprehensive survey of LLM-integrated ABMS systems. The authors categorize applications across social, physical, cyber, and hybrid domains, illustrating how LLMs extend the traditional ABMS pipeline by endowing agents with richer cognitive functions. They emphasize four major areas where LLMs significantly enhance agent design: perception, reasoning and decision-making, adaptation and memory, and heterogeneity. In LLM-based ABMS, agents are not limited to rule-based reactions; instead, they interpret environmental or text inputs in natural language, generate context-aware actions through chain-of-thought reasoning, reflect on past experiences to adapt over time, and express diverse personalities and communication styles.

Several representative frameworks, such as Generative Agents, CAMEL, and Voyager, demonstrate how LLM-powered agents can exhibit realistic social behavior, negotiation, and collaboration. These systems commonly integrate prompt templates, memory modules, reflection loops, and planner–executor architectures to simulate autonomous, interacting populations. Despite these innovations, Gao et al. identify several persistent limitations in the literature, including challenges in scaling simulations to large populations, the lack of standardized quantitative evaluation metrics, limited reproducibility, and ethical or alignment concerns arising from uncontrolled model behavior. Consequently, the current body of work establishes a strong architectural foundation but remains primarily qualitative, exploratory, and small-scale in nature.

SM3.3. Application Domain: Kudzu Invasion Modeling. First introduced from Japan, Kudzu is an aggressive and invasive species that causes ecological damage and reduces biodiversity. It costs billions of dollars annually to contain, maintain, and eradicate the species.

Aurambout & Endress (2018) [SM1] proposed a model to simulate the spread and management cost of Kudzu. They modeled the spread utilizing Cellular Automata on the NetLogo multi-agent programmable modeling environment. They ran many experiments utilizing different defined management policies to simulate the prevention/slowing of the Kudzu. After simulating the spread of Kudzu in each geographic patch, the cost of the management was calculated to determine the performance of the management policy. We could expand on their approach, utilizing their research on the behavior of Kudzu. However, we would implement an LLM-approach as a replacement for the management policies, analyzing the difference between hard defined policies between a flexible and dynamic LLM policy that will have potentially higher saved costs and reduced Kudzu spread.

Furthermore, Harron et al., 2020 [SM3] models the spread rate of Kudzu in the next five years using Monte Carlo simulation with data from Oklahoma, USA. The study also estimated the economic loss due to Kudzu using a replacement cost approach with a sensitivity analysis. For our project, this gives an additional method, Monte Carlo, to simulate the Kudzu spread other than the Cellular Automata method described before.

SM3.4. Intelligent Conservation Management. Silvestro et al. (2022) [SM5] demonstrates how a monitored, actuated loop can outperform static plans. Their method, CAPTAIN shows conservation as a monitor-decide-act loop optimized by RL under explicit budgets and unfolding over time, the cadence we'll mirror with monthly CA state reads followed by LLM management actions. Recurrent monitoring even with presence/absence noise improves outcomes, so our pipeline should accept low-fidelity surveillance and still update decisions each cycle. CAPTAIN also warns that choosing the wrong objective (e.g., maximizing area) can worsen biodiversity outcomes; for Kudzu we should reward spread suppression/eradication per dollar rather than acreage.

In Parker et al. (2003) [SM4], the hybrid multi-agent systems model and land-use/cover change (MAS/LUCC) framework is proposed as a way to couple a cellular landscape with agent-based decision makers, linked by explicit feedbacks between people and environment. This overcomes plain CA's limits, as CA can capture spatial diffusion/transition rules but cannot represent heterogeneous, decentralized human decisions or institutional behaviors that determine where, when, and how land is managed. MAS/LUCC adds agents (e.g., farmers, landowners, agencies) whose autonomous, interacting decisions operate on different patches over time, enabling complex social-spatial responses. The paper positions MAS/LUCC for both the explanation of social phenomenon and empirical policy analysis. It also offers a validation approach that combines the use of spatial/landscape metrics rather than totals alone, separate quantity accuracy (how much change) from location accuracy (where change occurs), and resolution-scale sensitivity tests to guard against nonlinearity and path dependence. For our Kudzu project, this directly motivates our hybrid design – CA for ecological spread with LLM-based management agents for human response.

SM3.5. Project Uniqueness. In order to make our project unique, we will connect the ideas from Kudzu research and the LLM-powered agent model. Currently, instead of the Monte Carlo simulation, we are thinking of connecting the ecological Cellular Automata model with the LLM-powered agent model. The connection would be a feedback loop that runs at each month or season.

SM4. Conceptual Model. Conceptually, our ecological model is a spatially explicit, stage-structured CA defined on a regular grid. Each cell in the grid represents a land parcel with a fixed land-cover type. We map NLCD land-cover categories to carrying capacities for kudzu: forests and similar natural covers have high capacities, agricultural areas have moderate capacities, developed areas have low capacities, and water and some urban types have effectively zero capacity. Within each cell we track four kudzu state variables corresponding to the life-cycle stages: the number of seeds in a seed bank, the number of seedlings, the number of saplings, and the number of adults. Time advances in weekly steps.

The within-cell dynamics follow a set of simple stochastic rules inspired by Aurambout and Endress. Each week, individuals in each stage experience mortality with a stage-specific probability. We use a binomial distribution using these probabilities to calculate how many individuals survive to the next time stamp. For example, if there are 10 adult Kudzu with a mortality rate of 0.01, the simulation runs a binomial trial with $n = 10$, $p = 0.99$ to determine how many remain. This simulates real world Kudzu death from events like disease, age, or other factors. At certain recruitment times of year, seeds may germinate into seedlings, seedlings may become saplings, and saplings may become adults, subject to the constraint that the total biomass in the cell cannot exceed its carrying capacity. These life stage promotion or maturation functions using binomial distributions through binomial trials similar to the previously mentioned mortality functionality since there is a chance that Kudzu won't mature. Adults also contribute to the seed bank at a rate that depends on their abundance described in a later paragraph. These processes capture the seasonal structure of kudzu growth: there are months where seedlings and saplings rapidly advance into larger, more impactful stages.

Spatial spread is modeled through two mechanisms. Local spread represents vegetative growth into neighboring cells. This is Kudzu's primary means of spread. We define a Moore neighborhood of eight neighboring cells around each cell, and at each weekly step, saplings and adults in a cell can colonize neighboring cells with a probability that increases with local density on a binomial distribution, again capped by the receiving cell's carrying capacity. This spread directly increases the sapling count in the cell and not seedlings because of the biological nature of asexual reproduction that Kudzu performs. On the other hand, long-distance dispersal represents rare events where actual seeds or propagules are transported farther afield by animals, humans, or water. In our conceptual model, each adult has a small probability of producing a small number of seeds based on a Poisson distribution which is ideal for a variable number of seeds. Then each seed travels from the origin parent to a location determined by an exponential distribution. This is because the seeds will realistically land close by. Once the distance has been calculated, each seed will be assigned a direction uniformly between 0π and 2π, since each direction is relative equal. Based on direction and distance, the new cell for seeds will be identified and the new cell will increase the amount of seeds that it currently has. The combination of local spread and LDD produces invasion patterns with expanding fronts and scattered satellite infestations.

The management agent interacts with this ecological process on a slower, monthly timescale. At the beginning of each month (after four or five weekly updates), we aggregate the CA state into a compact observation. This observation includes the current year and month, total counts in each life-cycle stage, the remaining budget and budget duration, and a ranked list of the worst hotspots defined by a weighted sum of adult, sapling, and seedling density. The observation also includes seed bank hotspots. The hotspots are not deterministic in order to simulate reporting of Kudzu which could be random and non-deterministic. The agent's action space consists of a small set of treatment types that can be applied to specific cells: generic treatments that proportionally reduce all stages, more targeted seed-bank removal that aggressively reduces seeds with some effect on other stages, and survey actions that spend a small amount of budget without changing the state but could, in extensions, reveal hidden infestations deterministically. The LLM commonly uses the survey action based on hotspot information to identify neighboring cells that need to also be treated or could potentially spread further. Each action type has an associated cost per cell. The agent's conceptual objective is to choose actions that reduce future kudzu spread and abundance while respecting a fixed total budget over the entire planning horizon.

SM5. Simulation Model. We implemented the conceptual CA in Python using a modular structure. The core ecological dynamics are encapsulated in a `KudzuField` class that loads the Clark County land-cover raster, maps each land-cover class to a carrying capacity, and initializes arrays for seeds, seedlings, saplings, and adults at every cell. Weekly updates apply mortality, recruitment, local spread, and long-distance dispersal using the parameters chosen from the literature and tuned for numerical stability. All counts are kept as integers, and we ensure that no cell ever exceeds its carrying capacity. A set of helper methods allows us to snapshot the current state, compute summaries such as total kudzu or hotspot locations, and apply treatment actions by decrementing counts in the targeted cells.

A separate `Runner` script orchestrates the interplay between the CA and the management policies. It seeds the landscape with an initial configuration consisting of four clusters of infested cells, mirroring the NetLogo model we used for inspiration. The runner then advances the simulation week by week. On weeks that correspond to the beginning of a new month, it constructs the observation summary, passes it to the selected policy, and receives back a list of actions. These actions are checked and priced by a `BudgetManager` object, which tracks remaining funds, computes the total cost of each proposed action list, and removes or scales actions that would exceed the budget. Valid actions are then applied to the `KudzuField` before the next block of weekly CA updates. Throughout the run, we log key metrics and write history files that record the state and management actions at each month.

The policies themselves are implemented in a separate module. We provide three non-LLM policies for comparison: a trivial no intervention policy that always returns an empty action list, a greedy policy that treats the single densest hotspot each month, and a simple heuristic policy that treats a fixed number of top hotspots at full intensity regardless of budget efficiency. The LLM policy constructs a detailed text prompt that describes the current observation, the definitions and costs of each action type, and instructions about budget discipline and formatting. It then calls a Gemini model through an OpenRouter-compatible API, expects a JSON-formatted list of actions, and parses that output. Basic error handling and retry logic are used to guard against malformed responses or temporary API failures. The policies and budget manager are written so that their interfaces match, allowing us to swap policies in and out while leaving the rest of the simulator unchanged.

We performed several verification steps to ensure that the simulation code matched our conceptual model. With management turned off and spread probabilities set to zero, we confirmed that the kudzu population declined over time purely due to mortality and that no new infestations appeared. With local spread turned on but long-distance dispersal disabled, we observed smooth expanding fronts around the initial clusters and no remote satellites, as expected. With both spread mechanisms active, the simulated invasion exhibited expanding clusters and scattered satellites similar to published figures. We also included consistency checks in the code to ensure that all counts remain non-negative and do not exceed cell capacities after each update, and we tested the budget manager with controlled action lists to verify that budget totals never became negative.

SM6. Experimental Results and Validation. To evaluate how different management strategies perform in our simulator, we designed a baseline experiment in which four policies manage the same kudzu invasion. In every run, the simulation horizon was 156 weeks, corresponding to three years. The landscape started with thirty infested cells arranged in several clusters in suitable habitat, and each policy was given a total budget of $30,000 to spend over the three-year period. Management decisions were made at the beginning of each simulated month. We fixed the random seed at 42 so that the underlying stochastic mortality, recruitment, and dispersal processes were identical across policies, ensuring that differences in outcomes are due to the management strategy rather than random variation.

The time-series trajectories show how the invasion unfolds under each strategy. Under no intervention, total kudzu grows steadily for roughly two years and then experiences large surges during recruitment periods, ending the simulation at a high level. The greedy policy, which treats only the single densest hotspot each month, stays very close to the no-intervention curve and even overshoots it at times, indicating that this naive strategy does not meaningfully slow the invasion. The simple heuristic policy, which treats multiple hotspots every month, holds the population to a much lower level than the first two policies but never eliminates it; the time series settles at a non-zero equilibrium. In contrast, the LLM policy quickly drives kudzu down: the total abundance peaks at a small value and then declines to zero, after which the system remains eradicated for the rest of the three-year period.

We summarize the performance of each policy using several metrics: peak total kudzu observed at any time, final total kudzu at week 156, the number of distinct cells that were ever infested (a measure of spatial spread), the total dollars spent, and the number of months with at least one management action. Under no intervention, the invasion reaches a peak of about 3,697 individuals and ends with approximately 3,498 individuals spread across 968 cells, with zero cost. The greedy policy spends $3,900 but performs slightly worse, with a peak of 3,910, a final total of 3,719, and 996 cells ever infested. The simple heuristic policy is far more aggressive: it reduces the peak to roughly 948 and the final total to 927 individuals, and it limits spread to 308 cells, but it spends $19,500 and still fails to eradicate kudzu. The LLM policy is the only one that achieves eradication in this scenario. It reaches a peak of just 64 individuals, ends with zero individuals, and allows infestation in only 32 cells total. It spends about $16,360, leaving a substantial fraction of the budget unused, and it intervenes in only nine months over the three years.

The bar-chart comparison highlights these trade-offs visually. The no-intervention and greedy policies occupy the high infestation, low cost corner: they are cheap but ineffective. The simple heuristic sits in a moderate infestation, high cost region, reflecting that it spends

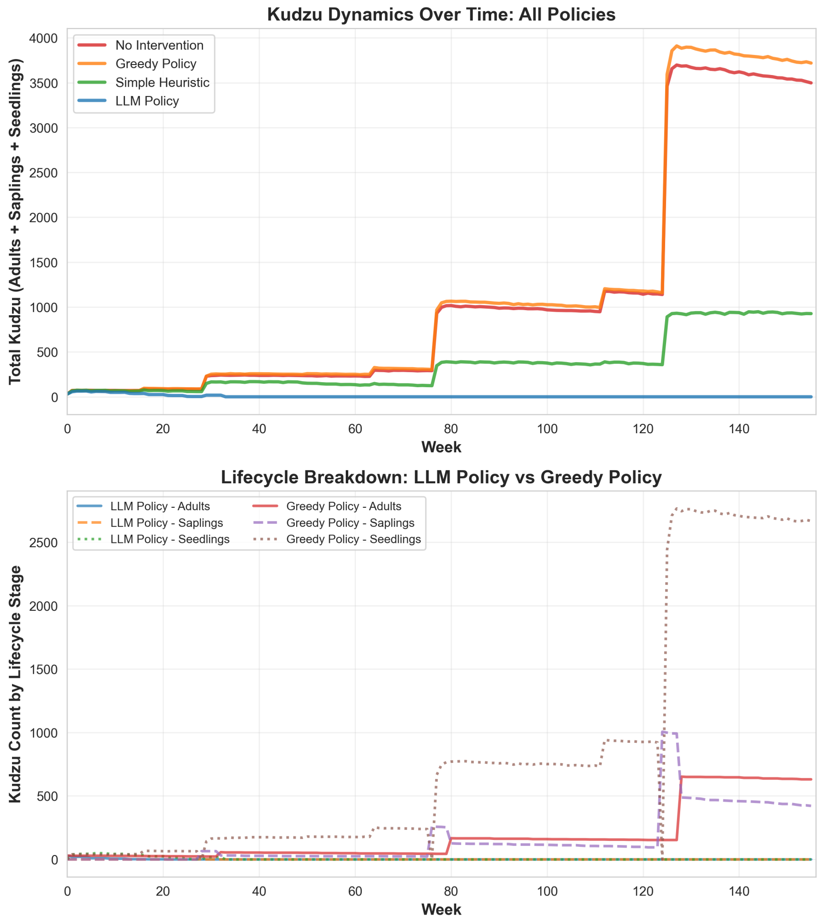

Figure SM1. *Kudzu over time by policy*

heavily to partially suppress the invasion. The LLM policy stands out by combining very low peak and final infestation and limited spatial spread with a mid-range cost and relatively few intervention months.

To examine cost-effectiveness more directly, we also compared final kudzu abundance against total budget spent. In this two-dimensional space, the most desirable region is low cost and low final infestation. No intervention appears at zero cost but high final abundance, and the greedy policy lies slightly to the right with essentially the same poor outcome. The simple heuristic moves toward the high-cost, moderate-abundance region: it buys some reduction at the expense of most of the budget. The LLM policy occupies the most favorable position, with moderate cost and zero final infestation.

Beyond these aggregate metrics, the policies differ in how they use the budget over time. The simple heuristic spends steadily throughout most of the three years, treating multiple hotspots almost every month and quickly consuming the majority of the available budget. The LLM policy instead front-loads its actions, focusing treatment and seed-bank removal in

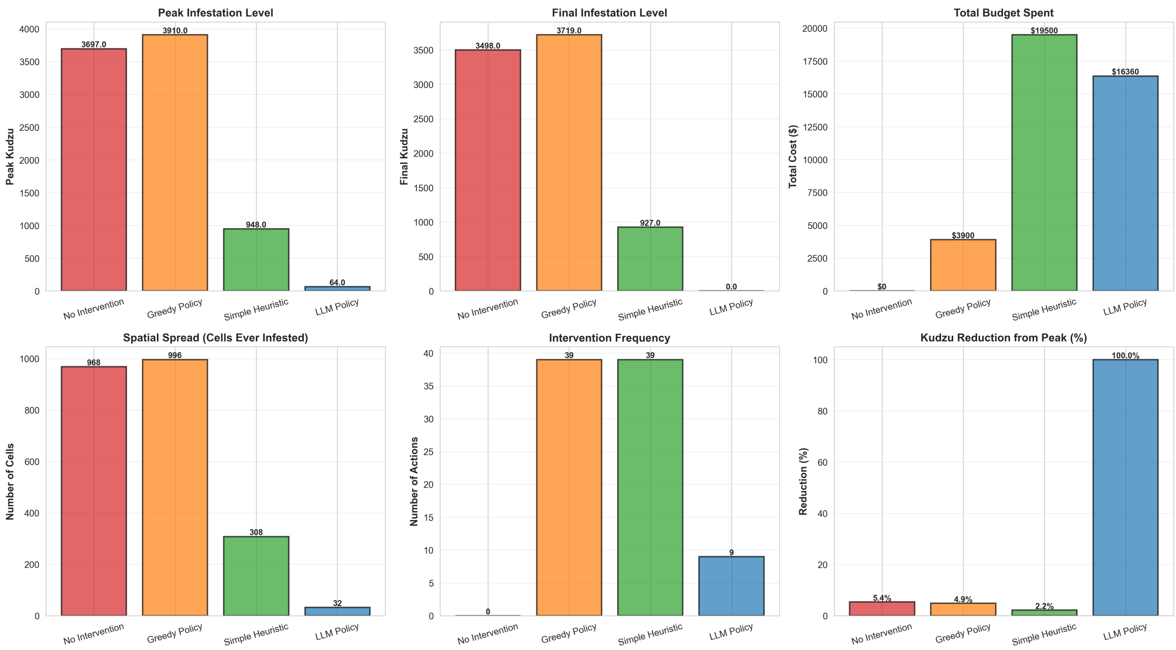

Figure SM2. *Policy comparison metrics*

the first one to one and a half years and then largely stopping once the invasion has been eradicated. This behavior aligns with the time-series results: early, concentrated intervention prevents the large outbreaks seen under the other strategies, which allows the LLM to save budget in the later years.

Our validation efforts focused on face validity and internal consistency rather than formal calibration to field data. Under no intervention and under the simple rules, the simulated invasion behaves qualitatively as expected: infestations expand outward from initial clusters, preferentially occupy high-capacity land-cover types, and respond strongly to seasonal recruitment. All four policies share the same pre-management dynamics early in the run, which provides a check that the underlying stochastic processes are consistent across runs. We also manually inspected samples of the LLM's proposed actions and their costs. Once given a sufficiently detailed prompt and clear constraints, the LLM tends to prioritize heavily infested clusters, uses seed-bank removal before key recruitment periods, and respects the budget limits enforced by the budget manager. However, we only ran a single random seed due to API and computational constraints, and we did not fit the model to real time-series data. For these reasons, the numerical performance gaps reported here should be interpreted as a case study rather than a statistically robust estimate. A more thorough validation, which we identify as future work, would involve multiple Monte Carlo runs per policy and comparisons against observed kudzu distributions.

SM7. Discussion, Conclusion, Summary. This project demonstrates how a classical ecological simulation framework and modern large language models can be combined to explore

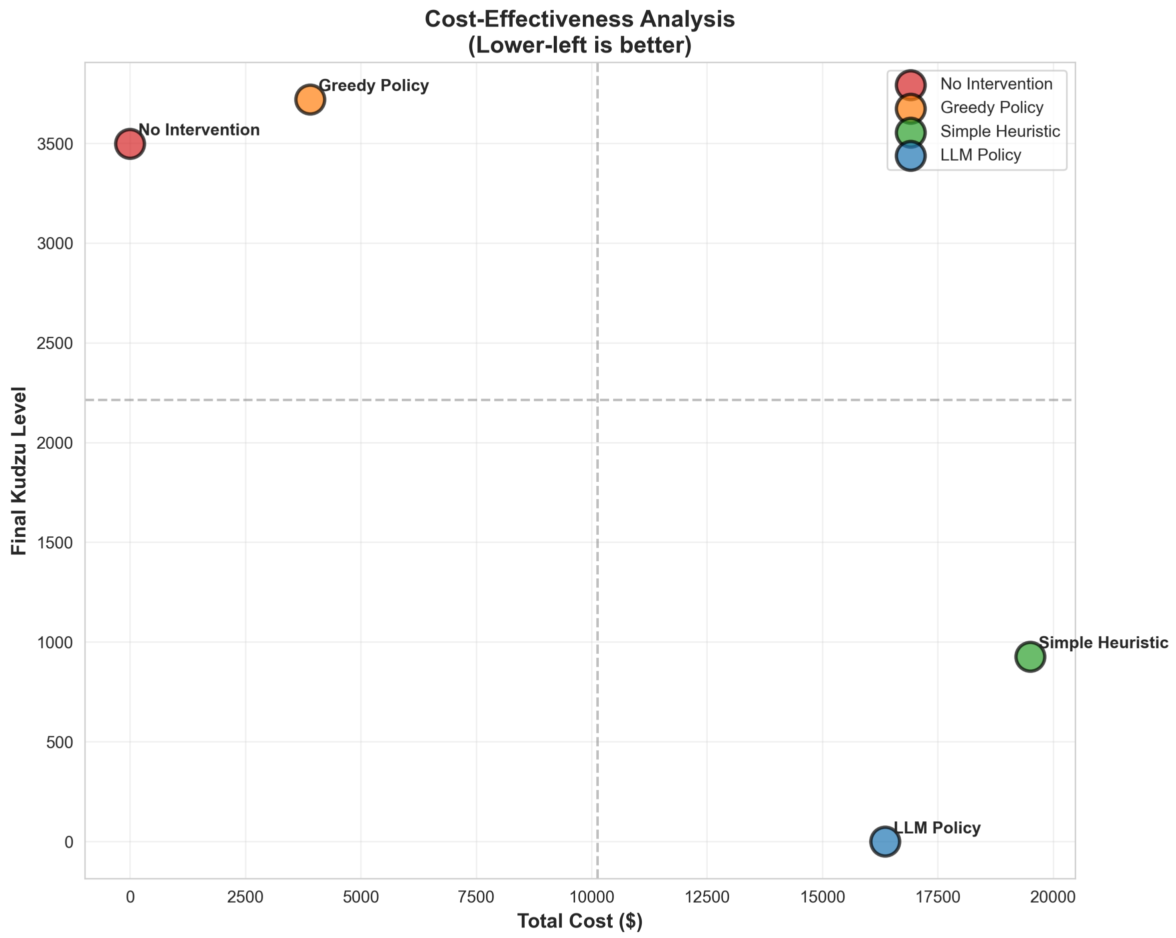

Figure SM3. *Cost vs final kudzu*

management strategies for invasive species. On the modeling side, we implemented a realistic CA of kudzu spread that respects land-cover heterogeneity, life-cycle structure, and both local and long-distance dispersal. On the decision-making side, we constrained a general-purpose LLM to a simple, well-defined action space and embedded it in a loop where it must respect budgets and react to partial summaries of the state. The resulting hybrid simulator provides a transparent environment in which LLM-based policies can be evaluated alongside simple rule-based baselines.

The main empirical finding from our baseline experiment is that the LLM policy substantially outperforms both the greedy and simple heuristic rules in this scenario. It eradicates kudzu within three years, uses fewer intervention months, and does not exhaust the available budget. The greedy policy under-reacts: by focusing only on the single densest cell each month and never addressing the seed bank, it allows the invasion to progress almost as if no management were occurring. The simple heuristic over-reacts: it spends heavily and treats many hotspots each month, limiting spread but never quite driving the system to zero. The LLM appears to exploit the temporal structure of the invasion, focusing on key clusters and life-cycle stages at the right times and then relaxing once the invasion is under control.

At the same time, the project highlights the limitations and challenges of using LLMs in this role. The quality of the LLM's behavior depends crucially on how we construct the prompt and observation; early versions that omitted important constraints produced unrealistic or budget-violating plans. API rate limits and query costs constrained the number of experiments and ablations we could perform. Our ecological model is also simplified: we ignore detection uncertainty, travel cost between sites, and heterogeneous stakeholder values, all of which matter in real management situations. Finally, because we relied on a single random seed and did not perform formal calibration against field data, our conclusions are based on one stylized case rather than a robust statistical ensemble.

If this work were extended, several directions seem most promising. First, repeated Monte Carlo experiments over different seeds and parameter sets would allow us to quantify variability and construct confidence intervals for metrics such as final infestation and total cost. Second, the action space could be enriched to include spatially contiguous treatments, different treatment intensities, or priority rules for protecting critical habitats, which would test whether the LLM can still find efficient policies when given more freedom. Third, the observation model could be made more realistic by introducing noisy or partial detection and by making surveys an important information gathering action rather than a placeholder. Finally, other decision-making agents such as reinforcement learning policies or planning algorithms could be implemented within the same framework and compared directly to the LLM, helping to clarify when language-based agents offer unique advantages.

Overall, our simulator shows that large language models can be integrated into ecological CA models as plausible adaptive managers, at least in simplified settings. The approach offers a flexible way to explore how high-level reasoning and domain knowledge encoded in LLMs might complement traditional optimization and rule-based strategies in conservation and invasive species management.

REFERENCES

[1] J. Aurambout and A. Endress, *A model to simulate the spread and management cost of kudzu (pueraria montana var. lobata) at landscape scale*, Ecological Informatics, 43 (2018), pp. 146–156, https://doi.org/10.1016/j.ecoinf.2017.12.004.

[2] C. Gao et al., *Large language models empowered agent-based modeling and simulation: A survey and perspectives*, Humanities and Social Sciences Communications, 11 (2024), https://doi.org/10.1057/s41599-024-03611-3.

[3] P. Harron et al., *Predicting kudzu (pueraria montana) spread and its economic impacts in timber industry: A case study from oklahoma*, PLOS ONE, 15 (2020), https://doi.org/10.1371/journal.pone.0229835.

[4] D. C. Parker, S. M. Manson, M. A. Janssen, M. J. Hoffmann, and P. Deadman, *Multi-agent systems for the simulation of land-use and land-cover change: A review*, Annals of the Association of American Geographers, 93 (2003), pp. 314–337, https://doi.org/10.1111/1467-8306.9302004.

[5] D. Silvestro, S. Goria, T. Sterner, and A. Antonelli, *Improving biodiversity protection through artificial intelligence*, Nature Sustainability, 5 (2022), pp. 415–424, https://doi.org/10.1038/s41893-022-00851-6.

[6] Z. Wu, R. Peng, T. Ito, and C. Xiao, *Llm-based social simulations require a boundary*, ArXiv, abs/2506.19806 (2025), https://api.semanticscholar.org/CorpusID:279999379.

SM8. Appendix A: Division of Labor. The work on this project was divided among the team members as follows. Aaron led the ecological modeling and CA implementation. He implemented the `KudzuField` class, defined land-cover-based carrying capacities, coded the life-cycle transitions and spatial spread mechanisms, and created the data parser that initializes infestations from the NetLogo-style data. Tristan focused on the integration between the CA and the management agent. He designed the structure of the monthly field observation, specified the action schema, implemented the code that applies actions to the CA at the beginning of each month, and helped wire these components into the main runner and experiment scripts.

Michael concentrated on the decision-making module and budget logic. He implemented the `llm_policy` function that constructs prompts and calls the Gemini API, designed and implemented the `BudgetManager` class that prices and validates actions, and modified the runners so that different policies could be selected via simple configuration flags. Linyuan led the analysis and visualization as well as bank seed removal action for the project. She wrote the scripts that run baseline comparisons, compute summary statistics from the JSON logs, and generate the full set of plots, including time-series comparisons, bar charts of key metrics, cost-effectiveness scatter plots, and the comprehensive dashboard figure. Jae initialized and maintained the GitHub repository, set up the branching structure, and coordinated code reviews and integration across branches. He also wrote and edited the LaTeX documentation for the checkpoint reports and this final report, making sure that the written description of the model, experiments, and results remained consistent with the evolving code.