Large Language Model-Assisted Workflows for Replicating Agent-Based Models

Boyu Wang[1][0000−0001−9879−2138] and Andrew Crooks[1][0000−0002−5034−6654]

Department of Geography
University at Buffalo, North Campus, Buffalo, NY 14260, USA
bwang44@buffalo.edu, atcrooks@buffalo.edu

Abstract. Replication is widely recognized as a cornerstone of scientific credibility, yet it remains relatively uncommon in agent-based modeling due to the substantial time, effort, and technical expertise required to reimplement models across platforms. This paper explores whether large language models (LLMs) can reduce these barriers and support the replication of agent-based models. We present two LLM-assisted workflows: a researcher-mediated approach that separates planning, evaluation, and implementation across specialized LLM roles, and a skill-assisted approach that combines reusable prompting strategies with an LLM-based coordination process. Using these workflows, we replicated three models in the Python-based Mesa framework. Validation of the replicated models employed structural, behavioral, visual, and statistical comparisons appropriate to each model. Across all three case studies, the Mesa implementations reproduced the key mechanisms, relationships, and emergent patterns reported in the original work. Where source code and statistical results were available, replicated outputs closely matched those from the original implementations. Our findings demonstrate that LLMs can substantially assist model replication from varying levels of source material. We conclude that LLM-assisted workflows offer a promising avenue for improving reproducibility, transparency, and knowledge transfer within the agent-based modeling community.

Keywords: Agent-Based Modeling · Generative AI · Large Language Models · Replication · Verification and Validation.

1 Introduction

Over the last several decades we have seen a proliferation of agent-based models being applied to a wide range of phenomena. While many challenges still exist with respect to the development of such models ranging from the construction of their behavioral rules to the validation of model outcomes. An often overlooked challenge is that of model replication [1, 14]. Replication has long been argued an important tenant and cornerstone in social simulation and sciences more generally and without it one cannot truly trust the results of such models [7]. However, replication is rarely done [1, 13, 29], instead modelers tend to build new models. This could be related to the fact that replication might not be as

rewarding as developing a new model [25], but if our simulations are to stand the test of time, replication is needed. Furthermore, successful replication allows for the development of new ideas and theories by altering the assumptions or scenarios of existing models [11], or carrying out more detailed analysis of model parameters than those reported in the original publication (e.g., [27]). For example [19] took an abstract model, replicated it and then and extended it with a realistic geographic population distribution.

Replication of agent-based models is not a new endeavor. For example in Axelrod [1] notes efforts to replicate eight well know models (e.g., Conway's game of life, Schelling's tipping model) in the 1990s. From which he noted four broad problem categories relating to replication. These ranged from ambiguity in published descriptions, gaps in published descriptions, to situations where published descriptions were clear but wrong. While the final category related to difficulties in the accessing source code of the models (e.g., only available on paper). Even when the source code is available, replication in another language can be challenging resulting in the replicators often contacting the original authors for clarifications (e.g., [25, 23]). For example, when Janssen [15] was replicating of the Artificial Anasazi model [2], he still needed assistance from the original programmer.

There have been numerous successful replications (e.g., [27]) , For example, North and Macal [20] were able to replicate the Beer Game supply chain model in several different modeling platforms (e.g., Repast and Swarm) and commented that the original model was robust in different implementations but they also noted "the model reproduction processes were challenging and time-consuming" [20]. Similar issues of the time consuming nature of replication have been made elsewhere (e.g., [25]), For example in [3], it was noted it took 60 hours to replicate two models. While in an attempt to replicate 18 simple models Grimm et al., [11] found that time it took to replicate a model varied between 2 and 12 days with an average being 5 days, where researchers worked 8 hours a day. Part of the time consuming part, is the implementation of code, for example when [4] replicated the Sugarscape model in MASON, approximately 3500 line of code were needed to be written.

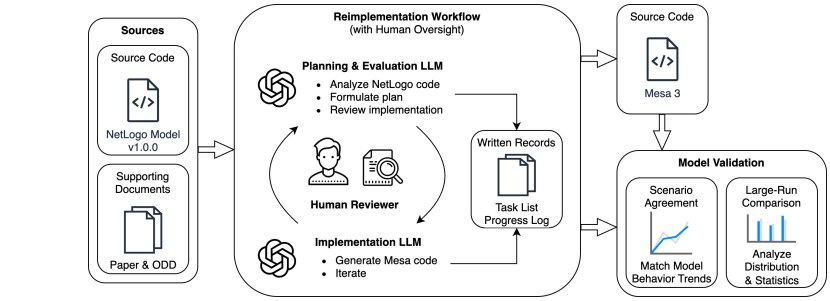
However, not all replication attempts have been successful (see [31, 24, 6]). This is especially the case when no source code is provided. For example in one such failed attempt, Will and Hegselmann [31] suggested the failure was due to unpublished hidden parameters which resulted in a heated debate between them and the original authors but it did result in the provision of the source code [18] and subsequent replication [30] which suggested that the results reported in the original paper were due to a "programming error". This is just one example of the value of replication, revealing shortcomings in the original published model with respect to model mechanisms. Others have noted reported model results are sensitive to specific parameter configurations, timescales the model was run for or to the number of agents simulated, thus the results reported were based on implementation rather than full model evaluation [7].

In light of the ongoing difficulties in replicating agent-based models, efforts have been made to aid model reimplementation, such as the development of the overview, design concepts, and details (ODD) protocol [12], documenting and tracing the model development cycle [10] or repositories for sharing the source code of models (e.g., `https://www.comses.net/`). Journals such as the Journal of Artificial Societies and Social Simulation are also encouraging code sharing and more detailed model descriptions rather than statements such as the "model is available upon request". However, sharing of code and providing ODD documentation is still not the norm. We believe that researchers should make greater efforts with respect to code sharing, model documentation, and linking code to documentation, as others have called for (e.g., [24, 1, 12]).

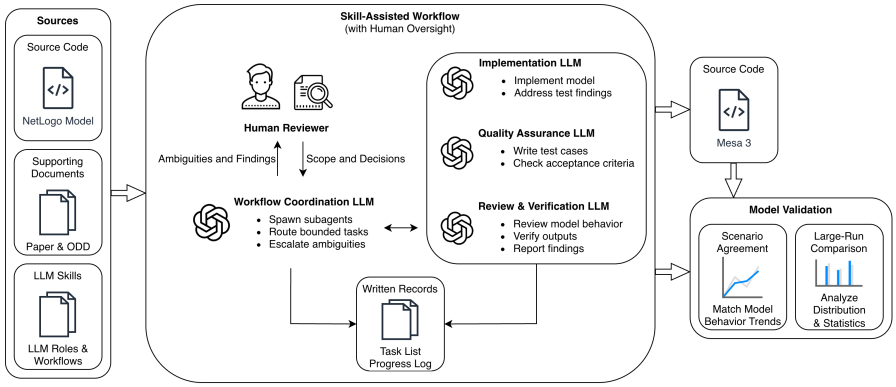
Building off the challenges of replication, one question arises, which is to what extent can generative artificial intelligence (AI), specifically large language models (LLMs) help with replicating agent-based models. Past work has shown LLMs can generate code and implement agent-based models from text prompts [5]. In addition to this we are seeing a significant amount of interest in empowering agent behaviors with LLMs [9] or utilizing for the creation of model documentation (e.g., `https://github.com/comses/odder`) However, we are seeing little in the use of LLMs in the area to model replication. One notable study has already demonstrated the different LLMs (e.g., Gemma, Claude ChatGPT) can generate python models from ODD documents and validate them against and existing NetLogo model, in this case it was a simple predator prey model [8]. Our paper differs from this, in the sense, we do not just look at one model but different types of models, some with ODD, some with code and one just the published paper. In the remainder of the paper, we present framework for model replication (in Section 2). Section 3 will show a series of replication experiments results, while Section 4 provides a summary of the paper and discusses avenues for future research.

2 Methodology

Three agent-based models were selected to be replicated in Mesa [16] as case studies where they differ in source materials and in how they could be validated (e.g., through statistical testing and visual comparison). For example, Ya-TASERPS [17] provided executable NetLogo code, model documentation, and published statistical findings. Sakoda's [26] checkerboard model was reconstructured from rules and examples in the original paper, whereas Slumulation [22] replication used its paper and NetLogo code. We examined two forms of role-separated LLM assistance to replicate these models: a researcher-mediated workflow for Ya-TASERPS (Figure 1A) and a skill-assisted workflow for Sakoda and Slumulation (Figure 1B). In both workflows, paper and other supporting documents such as the ODD protocol clarified definitions and design choices, and provided findings for comparison. When available, NetLogo code was re-run to generate expected model behavior and large-run statistics.



(a) Researcher-mediated workflow



(b) Skill-assisted workflow

Fig. 1. Two role-separated LLM-assisted workflows used in this study.

In the researcher-mediated workflow (Figure 1A), two LLMs were assigned distinct roles, with one LLM for planning and evaluation, and the other for source code implementation. Planning and evaluation LLM inspected the source materials, divided overall replication and validation tasks into phases, specified acceptance criteria (e.g., if the replications shared the same spatial configuration of the original model), and assessed completed work based on these criteria. Implementation LLM translated each phase into target source code and addressed evaluation findings. A human researcher oversaw their work and adjudicated ambiguities (e.g., discrepancies between the original model and replication). Intermediary outputs from both LLMs such as task list, progress log, findings and design decisions were kept as written records in separate files.

In the skill-assisted workflow (Figure 1B), LLMs were also tasked with their own designated roles. In addition to source materials such as NetLogo code and paper, LLM skills were also supplied as input, where researcher pre-defined reusable prompts specifying planning, implementation, and verification procedures for the LLMs. Inspired by an emerging practice *loop engineering* [21], in contrast to the researcher-mediated workflow, a workflow coordination LLM was introduced to spawn other LLMs and assign tasks. When ambiguities arose, the coordination LLM escalated them to the human researcher for resolution.

2.1 Example Models

The selected models differ in their available sources, model structure, and what could be validated. Each Mesa implementation was assessed by methods suited to its sources, from qualitative evidence of source-described patterns to comparisons with published statistical results.

Ya-TASERPS [17]. Ya-TASERPS is a youth-focused ABM grounded in interactional theory of delinquency [28]. In this study, its `v1.0.0` release developed using NetLogo is treated as the canonical behavioral source [17], with the paper and ODD as supporting documents. The model uses spatial bands for home, neighborhood, and school. Its daily schedule moves youths through school and neighborhood before evening family influence, and this ordering determines when the at-home update rule applies. The available sources allowed three forms of assessments: structural and mechanism checks, directional changes under specific parameter changes, and repeated-run comparisons with published findings.

Sakoda checkerboard model [26]. Sakoda's historical publication simulates the emergence of various social structures through the interactions between two social groups. We chose this model as there are few code examples that replicate this model. Furthermore, this is a good example of conceptional model, a textual description of a phenomena that is not executable, in the sense that no code is given which would be considered as an implemented model [29]. Furthermore, we were not involved with the development of the model - this replication is the creation of a new implementation. Output model code was generated solely based on the original paper itself with its eight published scenarios and their rule

sets: Crossroads, Mutual Suspicion, Segregation, Social Climber, Social Worker, Boy-Girl, Couples, and Husbands and Wives. Model validation inspected board patterns and supporting measures, with no behavioral or statistical comparisons.

Slumulation [22]. Slumulation represents informal settlement growth on a 51×51 land grid divided into nine wards. The paper describes the model, simulation experiments, and reported findings, while the executable NetLogo code provided detailed mechanics and experiment settings used to construct and assess the Mesa implementation. In Slumulation, households settle according to income, rent, availability, and co-location. Each annual step updates household states, patch rents and capacity, developer activity, city income, and summary measures before advancing simulation time; a patch becomes a slum when its occupants exceed its available housing units. The assessment of the Mesa replication consists of structural and mechanism checks (e.g., scheduler, settlement rules, rent updates, and slum definition), baseline comparison on NetLogo and Mesa trajectories and output distributions under the parameters used for the paper's typical run scenario, and three-way repeated-run comparisons between output statistics (i.e., mean and standard deviations of output parameters) from the paper, NetLogo model, and Mesa replication.

3 Results

In order to explore how well our workflows replicated the models we took several approaches, as noted by Axelrod [1] there are several ways to validate a replication. One such way is exploring the "relational equivalence" where the replication captures the same internal relationships as the original model and generates similar results. For interested readers, the three Mesa implementations and supporting materials are available at `https://github.com/wang-boyu/csssa2026-abm-replication`.

3.1 Ya-TASERPS

The Ya-TASERPS replication retained the geometry and population structure of the NetLogo model: bounds $x \in [-16, 16]$ and $y \in [-20, 20]$; 165 home, 1,023 neighborhood, and 165 school patches; and 150 youths. Family, individual, school, and neighborhood risk and promotive values were initialized within $[0.01, 0.99]$. The daily order remained `school` $\rightarrow$ `after_school` $\rightarrow$ `evening` $\rightarrow$ `step_day`. A youth away from home at the start of evening was moved home on the first substep, with the at-home update applying only on later evening substeps. These correspondences preserve the spatial structure and timing through which the tested risk and promotive processes operate.

Targeted checks covered 60 simulated days and input levels 25, 50, and 75 in balanced, risk-heavy, promotive-heavy, school-shift, and neighborhood-shift settings. Risk-heavy settings increased antisocial outcomes and reduced prosocial outcomes, whereas promotive-heavy settings produced the opposite pattern.

School-only and neighborhood-only shifts also changed outcomes in the directions consistent with the model's mechanisms. The Mesa replication therefore retained the intended relationships between environmental risk, promotive influences, and youth outcomes across the tested settings.

The repeated-run comparisons followed the same experimental design in the original Ya-TASERPS paper [17]. All combinations of six inputs - family, school, and neighborhood risk and promotive levels - were evaluated at three levels (0.25, 0.50, and 0.75), giving a full factorial design of $3^6 = 729$ parameter bundles. Ten runs of 1,800 simulated days were conducted per bundle. Across six prosocial and six antisocial input comparisons, NetLogo model and Mesa replication produced similar distributions. Figure 2 shows comparable distribution shapes for the tested input levels. The convergence of mechanism checks, targeted parameter responses, and large-run comparisons indicates that the Mesa replication preserves the risk and promotive relationships that Ya-TASERPS was designed to examine. It provides a basis for future experiments over the parameter space. The results support structural, directional, and repeated-run response correspondence over the tested experiments rather than exact numerical identity.

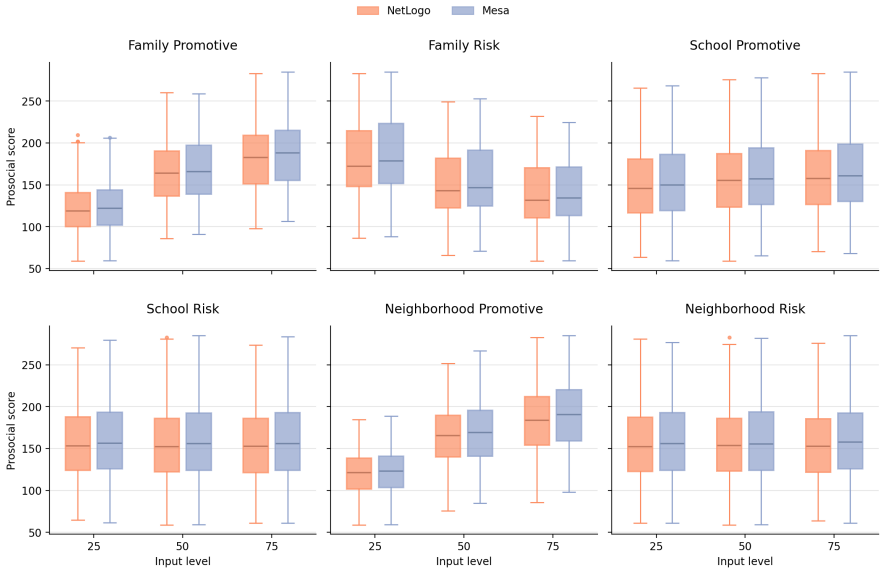
3.2 Sakoda

Across all eight scenarios from Sakoda's original paper, introduced in Section 2.1, the Mesa implementation produced resulting spatial patterns qualitatively consistent with those described and illustrated by Sakoda. The model used an 8×8 board with six Squares and six Crosses, the published attitude matrices, and distance weight $w = 4$ [26]. Under the documented conventions, a piece considered a distance-two jump when no adjacent move improved on staying, and all pieces were shuffled into one movement order per cycle. Five runs were examined for each scenario with visual assessment of the resulting board patterns.

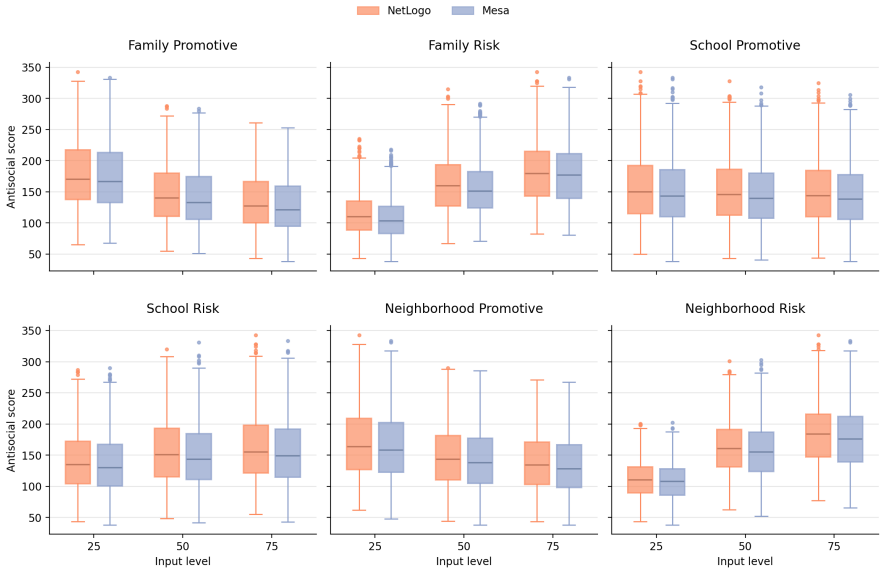
These patterns reflected the different social attitude matrices assigned to the scenarios. For instance, Crossroads formed compact, side-by-side same-group clusters near the center; Mutual Suspicion separated the groups; and Segregation produced cohesive groups on different parts of the board. Social Climber produced pursuit-and-escape organization, with persistent or recurrent movement in four or five runs. Figure 3 illustrates representative resulting states for these eight scenarios. These outcomes preserve the model's central relationship between social attitudes and emergent spatial structures. The comparisons, however, focused on broad spatial structures rather than exact Squares and Crosses coordinates or cycle-by-cycle trajectory. No numerical identity or statistical comparisons were formed or conducted.

3.3 Slumulation

The Slumulation replication reproduced the mechanisms and experimental structure described in the original paper and NetLogo code [22]. It retained the yearly order in which new households were created and settled before updating households, patches, developers, city income, and so on. Patch rent diffused over the



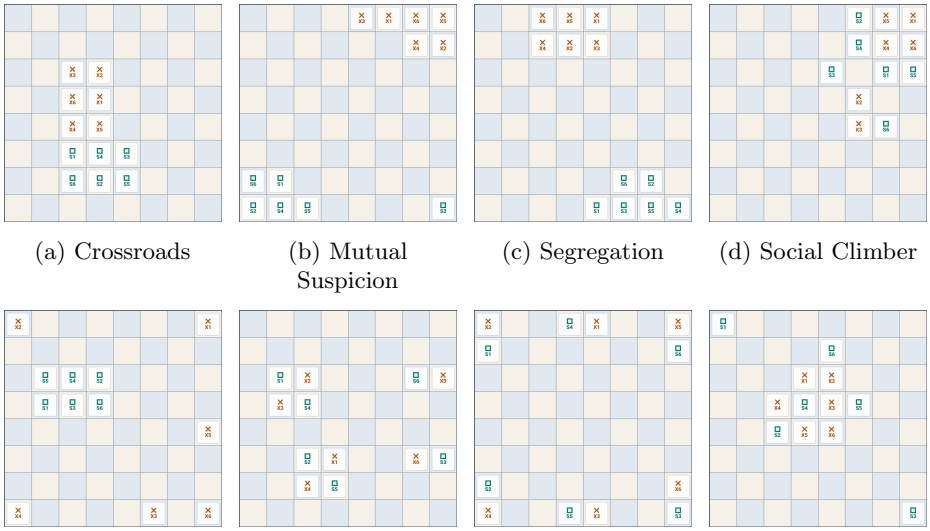
(a) Prosocial score



(b) Antisocial score

Fig. 2. NetLogo and Mesa distributions for (a) prosocial score and (b) antisocial score under various input levels.

wrapped eight-neighbor grid before rent growth, and occupancy relative to hous-



(a) Crossroads (b) Mutual Suspicion (c) Segregation (d) Social Climber

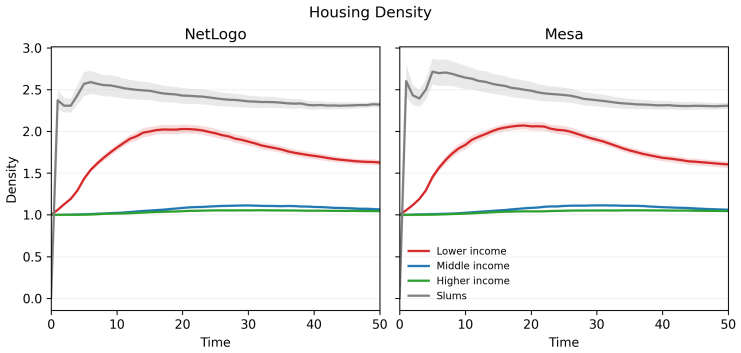
(e) Social Worker (f) Boy-Girl (g) Couples (h) Husbands and Wives

Fig. 3. Representative checkerboard states used as illustrations for comparison with source-described patterns.

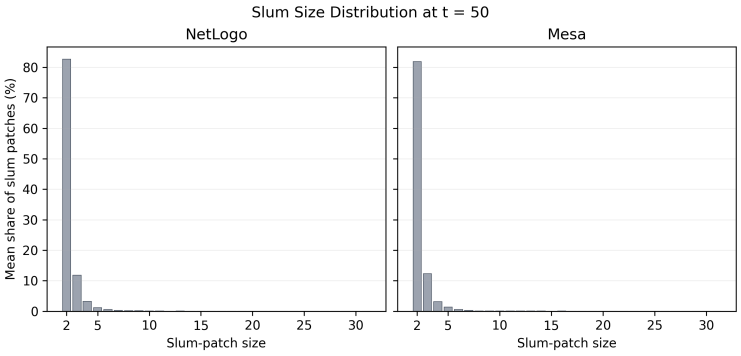
ing units determined whether a patch became a slum. These mechanisms provide the basis for comparing the spatial and aggregate dynamics of the NetLogo model and its Mesa replication.

To validate the replicated Mesa model, we conducted a three-way comparison of model outputs between the original paper, the release NetLogo model, and the Mesa replication. For the baseline scenario, we used the same input parameters as those in the typical simulation run from the original paper. Figure 4 (a) displays trajectories of slum, lower-income, middle-income, and higher-income housing density from 30 independent runs. Figure 4 (b) and (c) provide the slum-size distribution and slum population trajectories. Screenshots of an example baseline scenario at $t = 0$, $t = 25$, and $t = 50$ are shown in Figure 5. Under this baseline scenario, the original paper also reported statistics (mean and standard deviation) of key output parameters for 100 repeated runs. We collected the same statistics from 30 repeated runs of the NetLogo model and the Mesa replication, as reported in Table 1. Table 2 extends the comparison to four combinations of political intervention and developer activity settings. Across these experiments, NetLogo model and Mesa replication produced similar mean values to those reported in the paper with larger standard deviations, likely due to their smaller number of 30 runs.

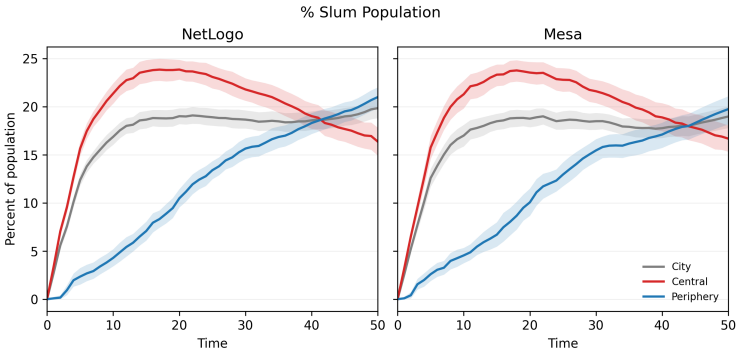
In addition to the baseline setting, the original paper conducted parameter sweeps to explore how model outputs could be affected by varying one parameter at a time, including population growth rate, economic growth rate, initial land supply conditions, and so on. The reported numbers were based on 100 re-



(a) Housing density



(b) Slum-size distribution



(c) Slum population

Fig. 4. Slumulation visualizations from 30 independent runs: (a) mean housing-density trajectories with 95% bootstrap intervals; (b) mean slum-patch-size distribution at $t = 50$; and (c) mean city, central, and peripheral slum-population trajectories with 95% bootstrap intervals.

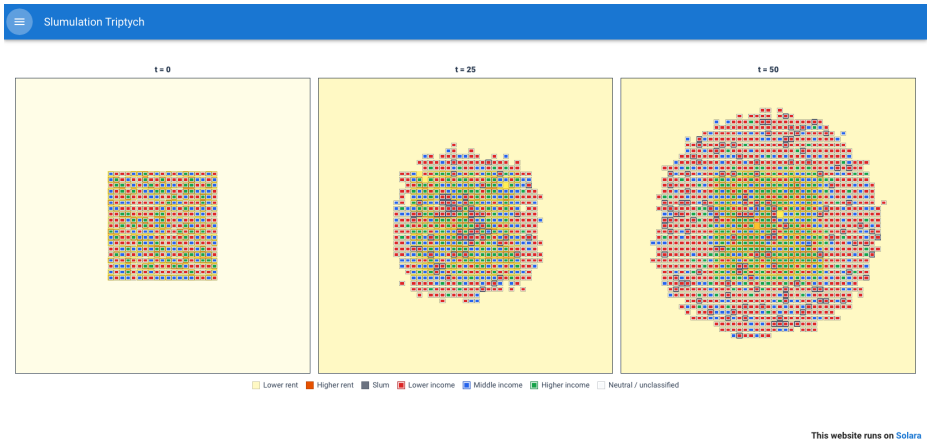


Fig. 5. Screenshots of a typical run, shown at $t = 0$, $t = 25$, and $t = 50$.

Table 1. Mean and standard deviation (in parentheses) of key output parameters at $t = 50$ from original paper (100 runs), NetLogo model (30 runs), and Mesa replication (30 runs).

Output	Area	Paper (100 runs)	NetLogo (30 runs)	Mesa (30 runs)
Slum population (%)	City	19.0 (0.3)	19.9 (2.8)	19.0 (3.2)
Slum population (%)	Center	16.8 (0.4)	16.4 (4.1)	16.7 (3.9)
Slum population (%)	Periphery	19.7 (0.4)	21.0 (3.1)	19.8 (4.1)
Area under slums (%)	City	10.1 (0.2)	10.6 (1.4)	10.1 (1.7)
Area under slums (%)	Center	8.5 (0.2)	8.3 (1.9)	8.4 (1.7)
Area under slums (%)	Periphery	10.6 (0.2)	11.3 (1.6)	10.7 (2.2)
Slum density	City	2.30 (0.01)	2.32 (0.10)	2.31 (0.10)
Slum density	Center	2.69 (0.04)	2.67 (0.37)	2.71 (0.35)
Slum density	Periphery	2.21 (0.01)	2.24 (0.08)	2.20 (0.08)
Number of slums	City	125.0 (2.0)	130.5 (15.3)	125.7 (19.3)
Number of slums	Center	24.0 (1.0)	23.9 (5.6)	24.1 (4.9)
Number of slums	Periphery	101.0 (2.0)	106.6 (13.7)	101.5 (19.7)

Table 2. Slumulation outputs at $t = 50$, shown as mean (standard deviation), with Paper, NetLogo, and Mesa values, showing how model outcomes can be affected by varying the Politics and Development parameters. P denotes political intervention and D developer activity; P on/D on is the baseline.

		P on / D on (baseline)			P off / D on			P on / D off			P off / D off		
Output	Area	Paper	NetLogo	Mesa	Paper	NetLogo	Mesa	Paper	NetLogo	Mesa	Paper	NetLogo	Mesa
Slum population (%)	City	19.2(0.3)	19.1(3.3)	19.3(2.4)	18.7(0.2)	18.6(3.0)	18.3(3.3)	24.4(0.4)	24.7(3.6)	24.2(3.8)	25.3(0.2)	25.6(2.2)	24.1(2.5)
Slum population (%)	Center	16.0(0.4)	16.3(4.0)	16.2(4.4)	13.9(0.3)	14.0(4.0)	14.8(3.5)	24.8(0.5)	23.6(5.6)	24.9(4.3)	22.0(0.4)	22.5(4.7)	20.1(4.6)
Slum population (%)	Periphery	20.3(0.3)	20.0(3.9)	20.4(2.8)	20.2(0.3)	20.2(3.5)	19.6(4.0)	24.2(0.4)	24.9(4.0)	23.9(4.2)	26.1(0.3)	26.4(2.3)	25.2(2.4)
Area under slums (%)	City	10.3(0.2)	10.2(1.8)	10.3(1.2)	10.1(0.1)	10.0(1.5)	9.8(1.7)	12.1(0.2)	12.2(1.7)	12.0(1.7)	12.5(0.1)	12.6(1.1)	12.1(1.2)
Area under slums (%)	Center	8.3(0.2)	8.2(2.1)	8.4(2.2)	7.3(0.1)	7.3(1.7)	7.4(1.5)	11.2(0.2)	10.6(2.3)	11.3(1.8)	9.6(0.2)	9.8(1.9)	9.1(1.8)
Area under slums (%)	Periphery	11.0(0.2)	10.8(2.1)	10.9(1.6)	11.0(0.2)	10.8(1.9)	10.6(2.2)	12.3(0.2)	12.7(2.0)	12.2(2.2)	13.3(0.1)	13.4(1.2)	12.9(1.3)
Slum density	City	2.26(0.01)	2.30(0.11)	2.30(0.12)	2.28(0.01)	2.29(0.11)	2.29(0.11)	2.35(0.01)	2.35(0.13)	2.34(0.14)	2.37(0.01)	2.39(0.09)	2.31(0.09)
Slum density	Center	2.54(0.04)	2.71(0.40)	2.62(0.32)	2.62(0.04)	2.59(0.45)	2.67(0.41)	2.62(0.03)	2.62(0.40)	2.64(0.38)	2.66(0.03)	2.68(0.39)	2.52(0.34)
Slum density	Periphery	2.20(0.01)	2.21(0.08)	2.23(0.09)	2.20(0.01)	2.23(0.08)	2.20(0.09)	2.28(0.01)	2.29(0.10)	2.27(0.11)	2.31(0.01)	2.32(0.07)	2.27(0.07)
Number of slums	City	128.0(2.0)	126.4(19.6)	128.0(12.8)	126.0(2.0)	124.2(16.7)	122.2(19.0)	158.0(2.0)	160.0(18.2)	157.0(18.9)	163.0(1.0)	163.7(11.5)	159.4(13.6)
Number of slums	Center	24.0(1.0)	23.7(5.9)	24.2(6.4)	21.0(1.0)	20.9(5.1)	21.5(4.3)	32.0(1.0)	30.6(6.6)	32.5(5.2)	28.0(1.0)	28.3(5.4)	26.2(5.3)
Number of slums	Periphery	105.0(1.0)	102.6(18.1)	103.8(13.0)	105.0(2.0)	103.2(15.8)	100.7(18.9)	125.0(2.0)	129.4(17.3)	124.5(18.5)	135.0(1.0)	135.4(10.2)	133.2(11.2)

peated runs for 50 time periods for each simulation experiment. While standard deviations were omitted in the original paper, it stated that they were all within the 2% to 4% range of all parameters. As an illustration, we conducted one of such experiments with varying population growth rate for the NetLogo model and Mesa replication. As shown in Table 3, in all three sources, slum population, area, density, and number are generally increased as population growth rose city-wide and in peripheral areas. Central area responses were less monotonic with respect to population growth. These results support replication of the selected mechanisms and reported patterns for the baseline and the examined parameter experiments, indicating the findings in the original paper are not unique to one implementation or software platform.

4 Summary and Outlook

We started this paper discussing issues with respect to replication of agent-based models and how through replication (i.e., the creation of a new implementation) one can gain confidence in the results reported by the publication. Replicating the same conceptual model in another programming environment allows researchers to examine whether reported findings can be reproduced beyond one specific implementation. However, we also noted the challenges with respect to replicating models, especially with respect to time, coding, and model complexities.

In this study, we examined how LLM-assisted workflows can support the replication and validation of agent-based models when the availability of source materials differ. We presented two workflows: a researcher-mediated workflow for Ya-TASERPS model and a skill-assisted workflow for Sakoda and Slumulation models, which allows us to take advantage of Generative AI. Applying these workflows to papers, documentation, and executable NetLogo code produced three Mesa implementations under distinct source conditions. As such we show the potential of LLMs to aid in the replication of agent-based models either from the description of the model or when model code and documentation is available. Another novelty of this workflow is using the LLMs to check how the replication results match the original results reported in the paper thus gaining a certain level of model validation. While the source materials determined both what could be reasonably implemented and compared, across the example models, generated code was the starting point rather than the final result of replication. The connection between implementation and validation is one of the contributions of the workflows presented here.

However, as with all research, there are limitations with our approach. First, although we were able to replicate the models, we did not explore the full set of sensitivities of them due to computational constraints. Second, with respect to the models we replicated from code, we might be as Wilensky and Rand [29] noted also suffering from "groupthink" in the sense that we adapted everything in the original model, bugs and all (if they existed). To some extent we just copied the model (apart from the Sakoda example). Thus further work needs to

Table 3. Slumulation outputs across population growth rates at $t = 50$, with Paper, NetLogo, and Mesa values. The 3% population growth rate is the baseline. Paper reports means from 100 runs with no standard deviations. NetLogo and Mesa report mean (standard deviation) from 30 runs per configuration.

		2%			3% (baseline)			4%		
Output	Area	Paper	NetLogo	Mesa	Paper	NetLogo	Mesa	Paper	NetLogo	Mesa
Slum population (%)	City	13.0	12.7 (2.0)	12.5 (2.1)	19.0	18.3 (3.6)	19.0 (2.9)	27.9	27.3 (3.7)	27.4 (3.0)
Slum population (%)	Center	14.2	14.4 (3.7)	14.1 (3.3)	16.3	16.1 (3.1)	16.6 (3.4)	12.1	12.9 (4.5)	11.5 (4.1)
Slum population (%)	Periphery	12.1	11.3 (2.2)	11.3 (2.2)	20.0	19.1 (4.3)	19.9 (3.2)	30.5	29.7 (4.1)	30.0 (3.2)
Area under slums (%)	City	7.4	7.2 (1.0)	7.1 (1.0)	10.1	9.9 (1.8)	10.2 (1.4)	14.3	14.0 (1.9)	14.0 (1.4)
Area under slums (%)	Center	8.1	8.3 (1.9)	7.9 (1.5)	8.2	8.6 (1.7)	8.8 (1.5)	4.9	5.4 (1.6)	4.9 (1.5)
Area under slums (%)	Periphery	7.0	6.5 (1.3)	6.5 (1.3)	10.7	10.2 (2.3)	10.6 (1.7)	15.9	15.5 (2.2)	15.5 (1.7)
Slum density	City	2.24	2.22 (0.16)	2.24 (0.19)	2.30	2.27 (0.13)	2.28 (0.11)	2.44	2.42 (0.11)	2.44 (0.10)
Slum density	Center	2.43	2.39 (0.30)	2.45 (0.44)	2.68	2.54 (0.44)	2.55 (0.33)	3.12	3.00 (0.74)	2.97 (0.80)
Slum density	Periphery	2.09	2.07 (0.06)	2.08 (0.06)	2.21	2.20 (0.09)	2.22 (0.08)	2.41	2.39 (0.10)	2.41 (0.08)
Number of slums	City	54.0	52.9 (6.4)	51.7 (7.0)	126.0	122.8 (20.2)	127.1 (15.6)	283.0	279.9 (31.9)	278.9 (22.4)
Number of slums	Center	23.0	24.1 (5.5)	22.9 (4.3)	24.0	24.8 (4.9)	25.3 (4.3)	14.0	15.6 (4.5)	14.0 (4.2)
Number of slums	Periphery	31.0	28.8 (5.5)	28.8 (5.3)	102.0	98.0 (19.6)	101.8 (14.1)	269.0	264.3 (30.8)	264.9 (21.8)

be carried out with respect to testing the sensitivities of the models. Third, it would also be interesting to explore models created in different languages and platforms (e.g., Java, AnyLogic) and see how well LLMs replicate them. For example, it has been noted that subtle differences in coding styles (e.g., NetLogo and Repast) for the same model can generate different results [6]. Fourth, as this was our initial foray into the use of LLMs in model replication and the workflows presented above were under development, we did not record researcher effort or elapsed time, model and tool usage (including calls, tokens, and computational runtime), or monetary cost. This study therefore cannot determine whether LLM assistance reduced the time, labor, or cost of replication but future work should explore this. Lastly, we also chose relatively simple models that did not utilize much in the way of external data, and one could argue it is easier to replicate simple models compared to more complex ones, [1], so again it would be interesting to see how well we could replicate models which utilize data such as GIS. With these limitations in mind, we see a lot of promise in the workflows presented here as an avenue for improving reproducibility, transparency, and knowledge transfer within the agent-based modeling community. Furthermore, there are many models created in NetLogo. As the use of Python is increasing, expanding libraries of example models developed with frameworks such as Mesa and Repast4Py would provide more resources for people to learn from.

References

1. Axelrod, R.: Advancing the art of simulation in the social sciences. In: Conte, R., Hegselmann, R., Terna, P. (eds.) Simulating Social Phenomena, pp. 21–40. Springer, Berlin, Germany (1997). https://doi.org/10.1007/978-3-662-03366-1_2
2. Axtell, R., Epstein, J.M., Dean, J.S., Gumerman, G.J., Swedlund, A.C., Harburger, J., Chakravarty, S., Hammond, R., Parker, J., Parker, M.: Population growth and collapse in a multiagent model of the Kayenta Anasazi in Long House Valley. Proceedings of the National Academy of Sciences **99**(3), 7275–7279 (2002). https://doi.org/10.1073/pnas.092080799
3. Axtell, R., Axelrod, R., Epstein, J.M., Cohen, M.D.: Aligning simulation models: A case study and results. Computational & mathematical organization theory **1**(2), 123–141 (1996)
4. Bigbee, A., Cioffi-Revilla, C., Luke, S.: Replication of Sugarscape using MASON. In: Agent-Based Approaches in Economic and Social Complex Systems IV: Post-Proceedings of The AESCS International Workshop 2005. pp. 183–190. Springer (2007)
5. Crooks, A., Jiang, N., Wang, B.: Generative AI and urban modeling. Environment and Planning B: Urban Analytics and City Science **52**(6), 1277–1281 (2025). https://doi.org/10.1177/23998083251351500
6. Donkin, E., Dennis, P., Ustalakov, A., Warren, J., Clare, A.: Replicating complex agent based models, a formidable task. Environmental Modelling & Software **92**, 142–151 (2017). https://doi.org/10.1016/j.envsoft.2017.01.020
7. Edmonds, B., Hales, D.: Replication, Replication and Replication: Some Hard Lessons From Model Alignment. Journal of Artificial Societies and Social Simulation **6**(4), 11 (2003), `https://www.jasss.org/6/4/11.html`

8. Fachada, N., Fernandes, D., Fernandes, C.M., Matos-Carvalho, J.P.: Can large language models implement agent-based models? an ODD-based replication study. Ecological Modelling **517**, 111624 (2026). https://doi.org/10.1016/j.ecolmodel.2026.111624
9. Gao, C., Lan, X., Li, N., Yuan, Y., Ding, J., Zhou, Z., Xu, F., Li, Y.: Large language models empowered agent-based modeling and simulation: A survey and perspectives. Humanities and Social Sciences Communications **11**(1), 1–24 (2024)
10. Grimm, V., Augusiak, J., Focks, A., Frank, B.M., Gabsi, F., Johnston, A.S., Liu, C., Martin, B.T., Meli, M., Radchuk, V., et al.: Towards better modelling and decision support: Documenting model development, testing, and analysis using TRACE. Ecological modelling **280**, 129–139 (2014). https://doi.org/10.1016/j.ecolmodel.2014.01.018
11. Grimm, V., Berger, U., Calabrese, J.M., Cortés-Avizanda, A., Ferrer, J., Franz, M., Groeneveld, J., Hartig, F., Jakoby, O., Jovani, R., et al.: Using the ODD protocol and NetLogo to replicate agent-based models. Ecological Modelling **501**, 110967 (2025). https://doi.org/10.1016/j.ecolmodel.2024.110967
12. Grimm, V., Railsback, S.F., Vincenot, C.E., Berger, U., Gallagher, C., DeAngelis, D.L., Edmonds, B., Ge, J., Giske, J., Groeneveld, J., et al.: The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. Journal of Artificial Societies and Social Simulation **23**(2) (2020), `https://www.jasss.org/23/2/7.html`
13. Hales, D., Rouchier, J., Edmonds, B.: Model-to-model analysis. Journal of Artificial Societies and Social Simulation **6**(4) (2003), `https://www.jasss.org/6/4/5.html`
14. Heppenstall, A., Crooks, A., Malleson, N., Manley, E., Ge, J., Batty, M.: Future developments in geographical agent-based models: Challenges and opportunities. Geographical Analysis **53**(1), 76–91 (2021). https://doi.org/10.1111/gean.12267
15. Janssen, M.A.: Understanding Artificial Anasazi. Journal of Artificial Societies and Social Simulation **12**(4) (2009), `https://www.jasss.org/12/4/13.html`
16. Kazil, J., Masad, D., Crooks, A.: Utilizing python for agent-based modeling: The Mesa framework. In: Thomson, R., Bisgin, H., Dancy, C., Hyder, A., Hussain, M. (eds.) Social, Cultural, and Behavioral Modeling. pp. 308–317. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-61255-9_30
17. Lee, J., Crooks, A.T.: Youth and their artificial social environmental risk and promotive scores (Ya-TASERPS): An agent-based model of interactional theory of delinquency. Journal of Artificial Societies and Social Simulation **24**(4), 2 (2021). https://doi.org/10.18564/jasss.4660
18. Macy, M., Sato, Y.: Reply to Will and Hegselmann. Journal of Artificial Societies and Social Simulation **11**(4), 11 (2008), `https://www.jasss.org/11/4/11.html`
19. Miodownik, D., Cartrite, B., Bhavnani, R.: Between replication and docking: "Adaptive Agents, Political Institutions, and Civic Traditions" revisited. Journal of Artificial Societies and Social Simulation **13**(3), 1 (2010)
20. North, M.J., Macal, C.M.: Agent-based modelling and systems dynamics model reproduction. International Journal of Simulation and Process Modelling **5**(3), 256–271 (2009). https://doi.org/10.1504/IJSPM.2009.031099
21. Osmani, A.: Loop engineering. AddyOsmani.com (Jun 2026), `https://addyosmani.com/blog/loop-engineering/`, published 7 June 2026; accessed 14 July 2026
22. Patel, A., Crooks, A., Koizumi, N.: Slumulation: An agent-based modeling approach to slum formations. Journal of Artificial Societies and Social Simulation **15**(4), 2 (2012). https://doi.org/10.18564/jasss.2045

23. Polhill, J.G., Parker, D., Brown, D., Grimm, V.: Using the ODD protocol for describing three agent-based social simulation models of land-use change. Journal of Artificial Societies and Social Simulation **11**(2), 3 (2008), https://www.jasss.org/11/2/3.html
24. Rouchier, J.: Re-implementation of a multi-agent model aimed at sustaining experimental economic research: The case of simulations with emerging speculation. Journal of Artificial Societies and social simulation **6**(4) (2003), https://www.jasss.org/6/4/7.html
25. Rouchier, J., Cioffi-Revilla, C., Polhill, J.G., Takadama, K.: Progress in model-to-model analysis. Journal of Artificial Societies and Social Simulation **11**(2), 8 (2008), https://www.jasss.org/11/2/8.html
26. Sakoda, J.M.: The checkerboard model of social interaction. The Journal of Mathematical Sociology **1**(1), 119–132 (1971). https://doi.org/10.1080/0022250X.1971.9989791
27. Seagren, C.W.: A replication and analysis of Tiebout competition using an agent-based computational model. Social Science Computer Review **33**(2), 198–216 (2015)
28. Thornberry, T.P.: Toward an interactional theory of delinquency. Criminology **25**(4), 863–892 (1987). https://doi.org/10.1111/j.1745-9125.1987.tb00823.x
29. Wilensky, U., Rand, W.: Making models match: Replicating an agent-based model. Journal of Artificial Societies and Social Simulation **10**(4), 2 (2007), https://www.jasss.org/10/4/2.html
30. Will, O.: Resolving a replication that failed: News on the Macy & Sato model. Journal of Artificial Societies and Social Simulation **12**(4), 11 (2009), https://www.jasss.org/12/4/11.html
31. Will, O., Hegselmann, R.: A replication that failed on the computational model in 'Michael W. Macy and Yoshimichi Sato: Trust, Cooperation and Market Formation in the U.S. and Japan. Proceedings of the National Academy of Sciences, May 2002'. Journal of Artificial Societies and Social Simulation **11**(3), 3 (2008), https://www.jasss.org/11/3/3.html