UNIVERSITY OF THE PHILIPPINES

Bachelor of Science in Applied Physics

Clarence Ioakim T. Sy

Classroom learning dynamics using a cellular automata spatiotemporal model comparing peer instruction and traditional instruction

Thesis Adviser:

Johnrob Y. Bantang, Ph.D.

National Institute of Physics

University of the Philippines Diliman

Date of Submission:
July 2025

Thesis Classification:
F

National Institute of Physics
College of Science
University of the Philippines
Diliman, Quezon City

ENDORSEMENT

This is to certify that this thesis entitled **Classroom learning dynamics using a cellular automata spatiotemporal model comparing peer instruction and traditional instruction**, prepared and submitted by Clarence Ioakim T. Sy in partial fulfillment of the requirements for the degree of Bachelor of Science in Applied Physics, is hereby accepted.

JOHNROB Y. BANTANG, Ph.D.
Thesis Adviser

The National Institute of Physics endorses acceptance of this thesis in partial fulfillment of the requirements for the degree of Bachelor of Science in Applied Physics.

WILSON O. GARCIA, Ph.D.
Director
National Institute of Physics

To thinking complexly

ABSTRACT

CLASSROOM LEARNING DYNAMICS USING A CELLULAR AUTOMATA SPATIOTEMPORAL MODEL COMPARING PEER INSTRUCTION AND TRADITIONAL INSTRUCTION

Clarence Ioakim T. Sy
University of the Philippines (2025)

Adviser:
Johnrob Y. Bantang, Ph.D.

Taken from SPP-2024 paper Peer instruction has recently become one of the popular means of classroom instruction in Physics Education. Such educational setup must involve both physical interaction with things and actually doing some procedural steps mentally or physically. In this study, we investigate the effects of different seating arrangements on the students' learning efficiency in peer instruction by modeling the transfer of knowledge within the class as a probabilistic cellular automata model. We compared the efficiency of learning between the traditional learning model and the peer instruction model. We found that in square classrooms with different lengths $L \in \{32, 48, 64, 96, 128\}$, the inner corner seating arrangement performed the best among the peer instructions setups in terms of both the time t_{max} it takes for all the students to learn and the classroom's learning rate m. This result is different from a previous study, where they found that the outer corner seating arrangement performed the best. The difference stems from the simplifications made in this model that may not reflect real world factors. Our model uses binary values in an isotropic system and does not consider memory or unlearning. However, despite these simplifications, we found that in smaller classrooms with slow learners, peer instruction is more efficient compared to the traditional learning model, just as previous studies have suggested.

PACS: 01.40.Ha Learning in education, 05.45.Pq Numerical Simulations, 89.75.-k Complex systems

Table of Contents

List of Figures

Chapter 1

Peer Instruction and Traditional Models of Teaching

something

1.1 History of Peer Instruction

Something something Mazur something something

1.2 Difference of Peer Instruction from Traditional Models

1.2.1 Details of Peer Instruction

1.2.2 Benefits of Peer Instruction

1.2.3 Drawbacks of Peer Instruction

1.2.4 Existing models of Peer Instruction

- Nitta mathematical models of PI [4]
- roxas2010seating [5]

1.3 Problem statement

point is to propose an alternate model for a dynamical something

Chapter 2

The Classroom as a PCA

The classroom is a complex system that can be modeled as a probabilistic cellular automata (PCA). This chapter will discuss the classroom as a complex system and the probabilistic cellular automata model. The chapter will also discuss the implications of the model on the classroom and the teaching-learning process. (AI Generated Text)

Probably add objectives here???

2.1 Cellular Automata and its use in modelling complex systems

A two-dimensional (2D) rectangular cellular automata can be defined by a five-tuple [1] + Reinier MS:

$$\text{CA} = \{\mathcal{S}, \mathcal{C}, \mathcal{L}, \mathcal{N}, \mathcal{R}\} \tag{2.1}$$

where

$\mathcal{S} =$ is the set of possible states that each cell can assume. The state s can use any kind of representation such as the set of integers $\{0, \ldots, n-1\}$ with n as the total number of possible states.

$\mathcal{C} =$ $\{c = i, j \mid i \in \{1, 2, 3, \ldots, L_1\}, j \in \{1, 2, 3, \ldots, L_2\}$ s.t. $L_1 \times L_2 = N\}$ is the set of identifiers for each cell in the automaton where N is the total number of cells and L_1 and L_2 are the lengths of each side of the automaton space. The cells can then be identified by their position in the automaton (i, j). So, the state of cell $c \in \mathcal{C}$ can be written as $s_c = s_{i,j} \in \mathcal{S}$

$\mathcal{L} =$ defines the lattice neighborhood which is generally a mapping $f : \mathcal{C} \rightarrow C^M$ where M is the number of neighbors of a cell $c \in \mathcal{C}$. Any given cell c is mapped to another tuple of cells: $L_{i,j} = \{(i-1, j-1), (i-1, j), (i-1, j+1), (i, j-1), \ldots, (i, j)$. Where r is the radius of the Moore neighborhood. We then say that $\mathcal{L}_{i,j}$ contains the set of neighboring cell for $c_{i,j}$.

$\mathcal{N} = \mathcal{S}^M$, the set of neighborhood states. Thus, $N_c = N_{i,j} \in \mathcal{N}$ such that each $\mathcal{N}$ is in the form of the M-tuple $\{s_{i-1,j-1}, s_{i-1,j}, s_{i-1.j+1}, s_{i,j-1}, \ldots, s_{i,j}\}$.

$\mathcal{R} =$ defines the set of rules implemented in the CA with $g : s_{i,j} \mid \mathcal{L} \rightarrow \mathcal{S}$ as the mapping of any neighborhood state N_c to a new state $s'_{i,j}$ of the cell c. At the next time step, $s'_{i,j}$ replaces the original state $s_{i,j}$.

$\mathcal{N}$ can vary with the neighborhood structure and the boundary conditions of the automaton. The neighborhood structure dictates the shape the neighborhood in the lattice. Common neighborhood structures include the von Neumann (diamond) and Moore (square) neighborhoods. Boundary conditions dictate how the automaton treats cells at the egde of the lattice when determining the neighborhood. Common boundary conditions include toroidal, spherical, and fixed boundary conditions.

$\mathcal{R}$ can also be affected by other factors such as whether the rules are deterministic or probabilistic and whether they are implemented synchronously or asynchronously. An automaton with deterministic rules will always produce the same output given the same input, while an automaton with probabilistic rules will produce different outputs given the same input. In Conway's Game of Life, a cell dies when it has three live neighbors, while a cell is born when it has two or three live neighbors. This is an example of a deterministic rule. An example of a probabilistic rule would be a cell dying with a probability of 0.25 when it has three live neighbors. An automaton with synchronous rules will update all cells simultaneously, while an automaton with asynchronous rules will update cells one at a time. (something explanation something about sync vs async)

Due to the flexibility of cellular automata, they can be used to model a wide variety of complex systems. Cellular automata have been used to model physical systems such as fluid dynamics, biological systems such as the spread of diseases, and social systems such as traffic flow [3]. Its discreteness and locality make it a good model for systems that are composed of many interacting parts. Thus, we have chosen to use a two-state probability cellular automata to simulate the learning process for students in the classroom

2.2 PCA model for classroom dynamics

We used a two-dimensional binary probabilistic cellular automata (PCA) model to simulate the learning process in a classroom. In this PCA model, each cell in the automaton represents a student and the state of each cell represents their aptitude $S = \{\text{unlearned, learned}\} = \{0, 1\}$. We assign the neighborhood to be an outer-totalistic Moore neighborhood of radius $r = 1$ and define the boundary conditions to be fixed wherein the grid does not wrap around itself and $s_{i,j} = 0$ for $i, j \notin [1, L]$. How the aumaton updates varies on the learning set up of the classrom. In this study we tackle two learning set ups: traditional and peer instruction (PI). We take the traditional instruction to be the case where the teacher is the only source of information for the students - which is typical for classes where the time is mostly spent on lectures. In the PI set up, we consider that the students learn from each other but only when the students of interest already know the lesson or is in the learned state.

2.2.1 Traditional Instruction

To model the learning process in a classroom with the traditional instruction set up, we set the probability for the student to learn in each time P_{ij} to be dependent only on their individual learning rates $\lambda_{i,j}$, which describes how receptive they are to learning, and the probability for the student to learn from the teacher ρ_0. ρ_0 can be changed per individual based on where they are seated, but for simplicity, we set it to be the same for all students.

$$P_{ij} = \lambda_{ij}\rho_0 \tag{2.2}$$

where

$P_{i,j} \in [0, 1]$ is the probability of student $c_{i,j}$ to learn in each time step,

$\lambda_{i,j} = 1$ is the learning rate of student $c_{i,j}$

$\rho_0 \in [0, 1]$ is the probability of $c_{i,j}$ to learn from the teacher

In five-tuple form, the traditional PCA model for the classroom can be written as:

$\mathcal{S} =$ {learned, unlearned} $= \{0, 1\}$

$\mathcal{C} = \{(1,1),(1,2),\ldots,(1,L),(2,1),(2,2),\ldots,(2,L),\ldots,(L,L)\}$ where L is the length of the square classroom.

$\mathcal{L} = f(c) \leftarrow [L_c = \{(i+\delta i, j+\delta j)\ \forall\ (\delta i \wedge \delta j), \delta i, \delta j \in \{-1,0,1\}\}$ as a mapping for outer-totalistic Moore neighborhood of radius $r = 1$ with a fixed boundary condition.

$\mathcal{N} = \{00000000, 00000001, \ldots, 11111111\}$ such that the representation of the neighborhood state $N_c \in \mathcal{N}$ is equivalent to $N_c = \{s_{i+\delta i, j+\delta j} \forall \delta i, \delta j \in \{-1, 0, 1\}\}$.

$\mathcal{R} =$ the probabilistic rule defined by Equation 2.2.

The numerical procedure is outlined in Figure 2.1. Each simulation for the traditional model starts with all students unlearned. The simulation is considered finished once all the students have learned.

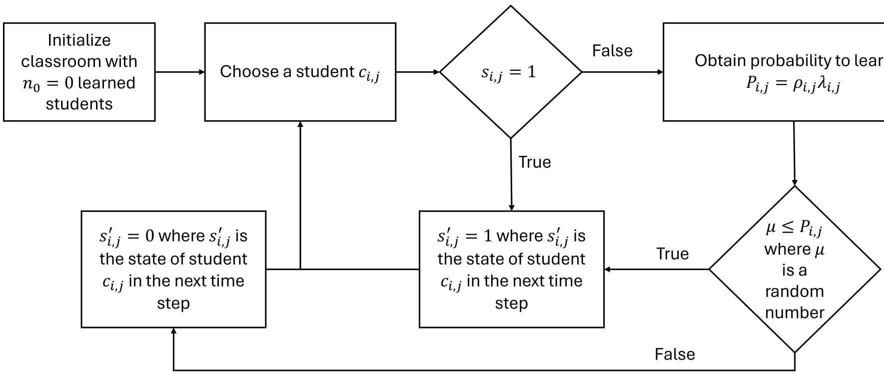
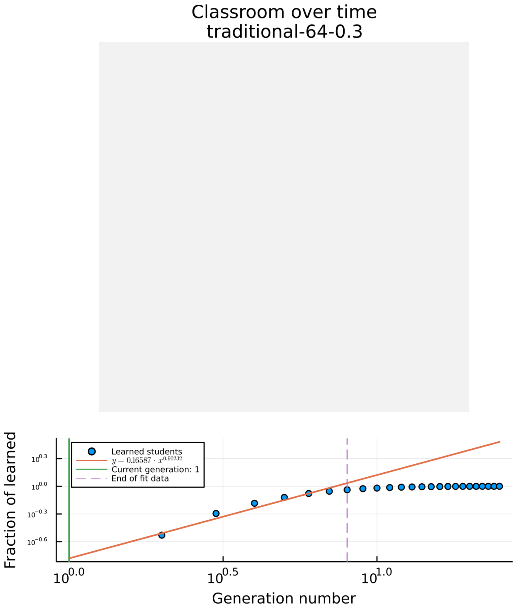


Figure 2.1: Numerical process for simulation of 2D BPCA for traditional set-ups.

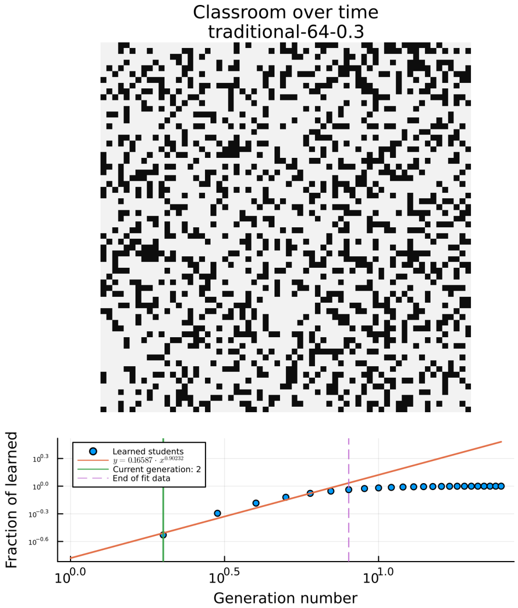
Figure 2.2 shows an example of a classroom's evolustion over time in the traditional set up. The sample classroom is set with $L = 64$, $\lambda_0 = 1$, and $\rho_0 = 0.5$.

2.2.2 Peer instruction (PI)

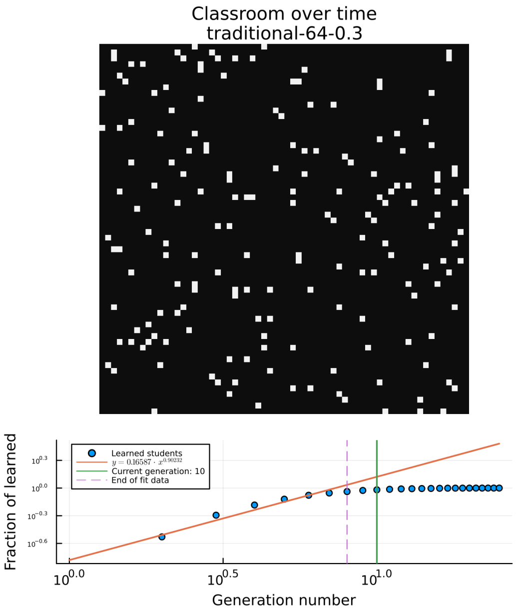
To model the learning process in a PI set up, we set the probability for the student to learn in each time P_{ij} to be dependent on three factors. First (1), their learning rate $\lambda_{i,j}$. Secondly (2), the positional learning factor $\rho_{i+\delta i, j+\delta j}$ which describes how likely it is to learn from the neighbor $c_{i+\delta i, j+\delta j}$ based solely on their relative position with



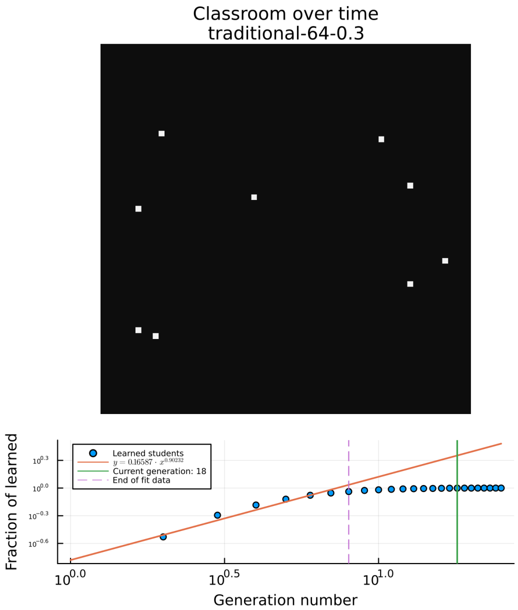
(a) $t = 1$



(b) $t = 2$



(c) $t = 10$



(d) $t = 18$

Figure 2.2: Sample classroom evolution for $L = 64$, $\lambda_0 = 1$, and $\rho_0 = 0.3$ with traditional instruction. Black squares represent learned students while white squares represent unlearned students. The accompanying graphs show the fraction of learned students as a function of time step. The blue circle represents the data points while the orange line shows the power law fit. The broken pink line shows the where we truncate the data for fitting the power law and the green line shows the current time step in the plot.

respect to $c_{i,j}$. Lastly (3), the aptitude level of the neighbor $s_{i+\delta i,j+\delta j}$ which dictates whether student $c_{i,j}$ can learn from them. The probability for a student to learn in each time step is then determined by the following equation:

$$P_{ij} = 1 - \prod_{\forall \delta i, \delta j} [1 - (\lambda_{ij})(\rho_{i+\delta i,j+\delta j})(s_{i+\delta i,j+\delta j})] \tag{2.3}$$

where

$P_{i,j} \in [0,1]$ is the probability of student $c_{i,j}$ to learn in each time step,

$\lambda_{i,j} = 1$ is the learning rate of student $c_{i,j}$

$\rho_{i+\delta i,j+\delta j} \in [0,1]$ is the probability of $c_{i,j}$ to learn from their neighbors in seats $\{c_{i+\delta i,j+\delta j} \forall \delta i, \delta j \in \{-1,0,1\}\}$ solely based from their relative positions with each other, and

$s_{i+\delta i,j+\delta j} = \{\text{unlearned, learned}\} = \{0,1\}$ are the neighbors' aptitude level.

The derivation of Equation 2.3 is shown in Appendix A.1.

The numerical procedure is outlined in Figure 2.3. Each simulation starts the classroom with four learned students $n_0 = 4$ placed in different seats in the classroom. The simulation is considered finished once all the students have learned.

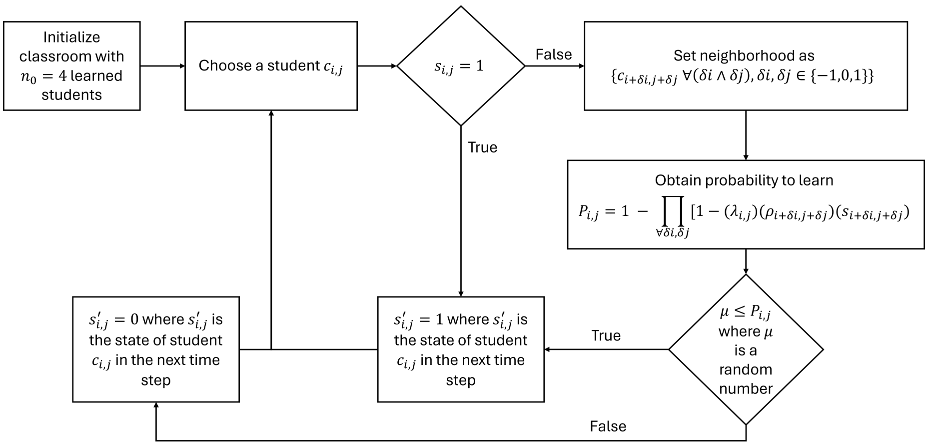


Figure 2.3: Numerical process for simulation of 2D BPCA for PI set-ups.

The seating arrangement (SA) were chosen from a previous study that shows that the SA can affect the learning process [5]. These SA's are namely: inner corner, outer corner, center, and random. The different SA's are shown in Figure 2.4.

Figure 2.5 shows an example of classroom's evolution over time. The sample classroom is set with $L = 64$, $\lambda_0 = 1$, and $\rho_0 = 0.5$ with the inner corner SA.

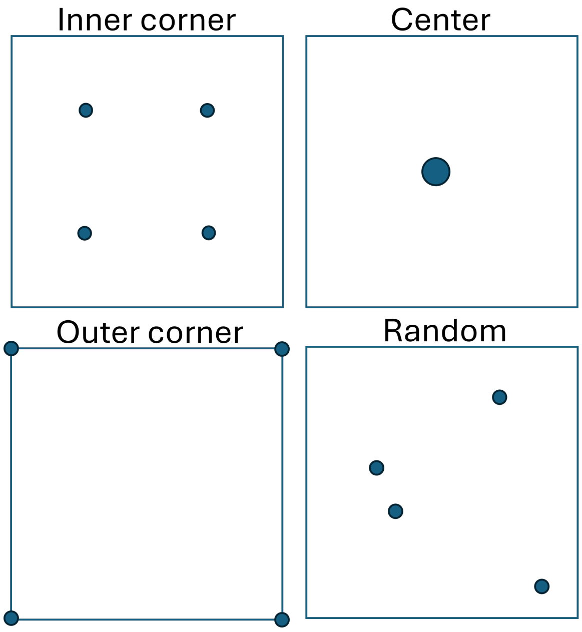
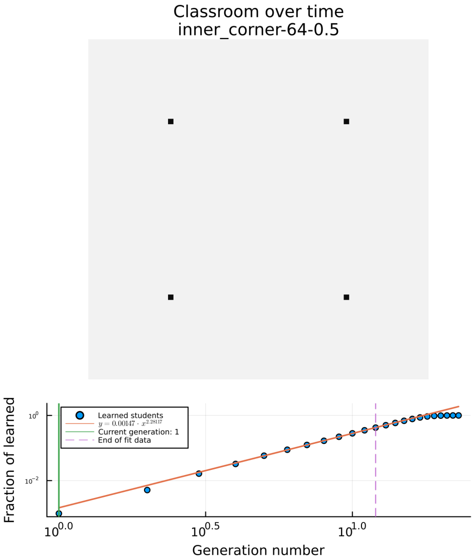
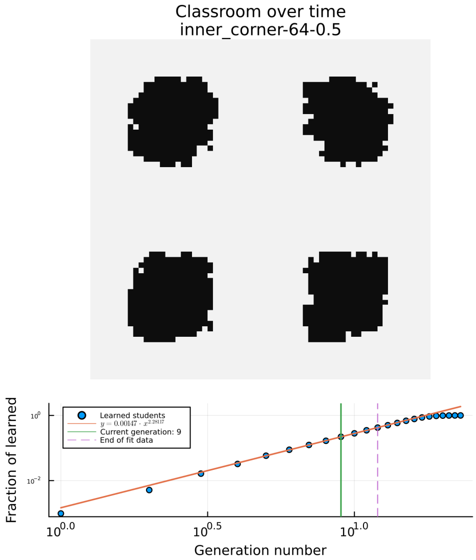


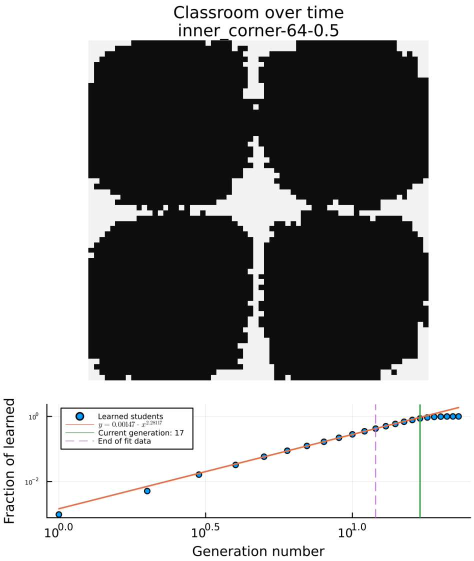
Figure 2.4: Peer instruction seating arrangements. Circles denote high aptitude students. The inner corner SA places high aptitude students halfway between the center and the corner of the classroom. The outer corner SA places high aptitude students at the corner of the classroom. The center SA places high aptitude students in the center of the classroom. The random SA places high aptitude students randomly throughout the classroom.



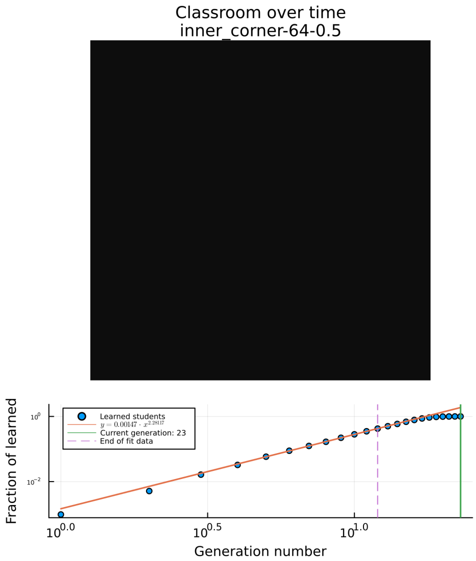
(a) $t = 1$



(b) $t = 9$



(c) $t = 17$



(d) $t = 23$

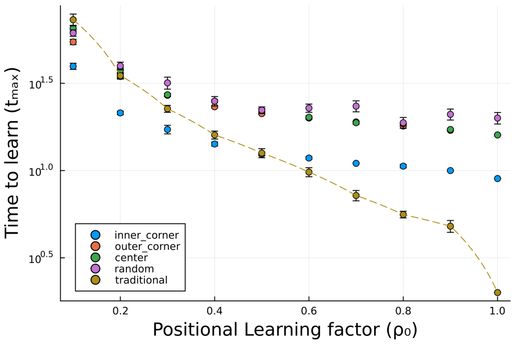
Figure 2.5: Sample classroom evolution for $L = 64$, $\lambda_0 = 1$, and $\rho_0 = 0.5$ with the inner corner SA. Black squares represent learned students while white squares represent unlearned students. The accompanying graphs show the fraction of learned students as a function of time step. The blue circle represents the data points while the orange line shows the power law fit. The broken pink line shows the where we truncate the data for fitting the power law and the green line shows the current time step in the plot.

2.3 Results: Homogenous PI vs Traditional

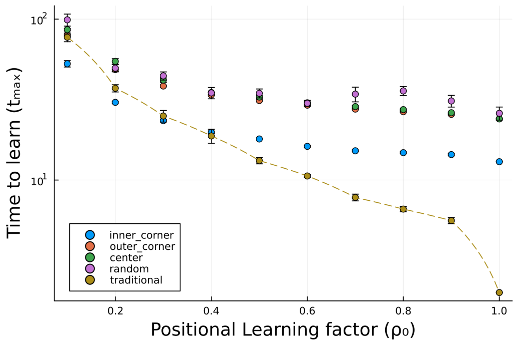
In this chapter, we only consider the case where all students have the same learning rate $\lambda_{i,j} = \lambda_0 = 1$ and an isotropic positional learning factor $\rho_{i,j} = \rho_0 \ \forall \ (\delta i \wedge \delta j), \delta i, \delta j \in \{-1, 0, 1\}$. From the simulations, we compared both the average number time steps $\langle t_{max} \rangle$ it takes for all the students in the classroom to learn and the average learning rate $\langle m \rangle$ across different configurations over 5 independent runs. The class learning rate m for each trial was obtained by using a Levenberg-Marquardt algorithm to fit a power law ($y = ax^m$) to the fraction of learned students as a function of time step. We only considered the first 50% of the data for the PI model and the first 25% of the data for the traditional model. This truncation was done so that we only fit the part of the data before the finite size effect starts to affect the simulation.

2.3.1 Time to learn t_{max} vs positional learning factor ρ_0

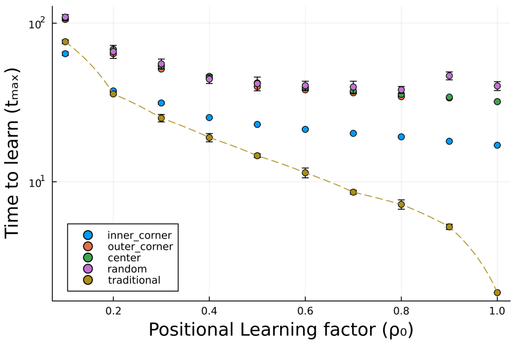
The data in Figure 2.6 shows that the time to learn t_{max} decreases with increasing positional learning factor ρ_0 for all classroom sizes for both traditional instruction and PI. Among PI SA's, the inner corner SA has the lowest t_{max} for all classroom sizes. The outer corner and center SAs performed similarly, while the random SA has the highest t_{max}. The traditional model performs situationally better than even the inner corner PI model, something we will investigate in Section 2.3.3.



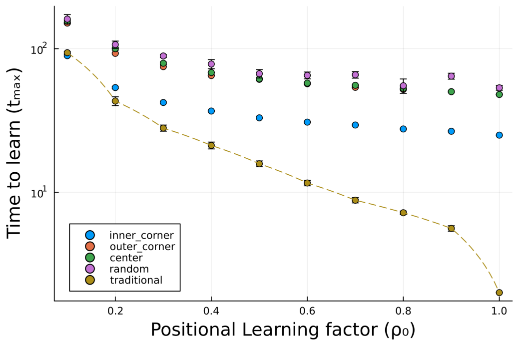
(a) $L = 32$



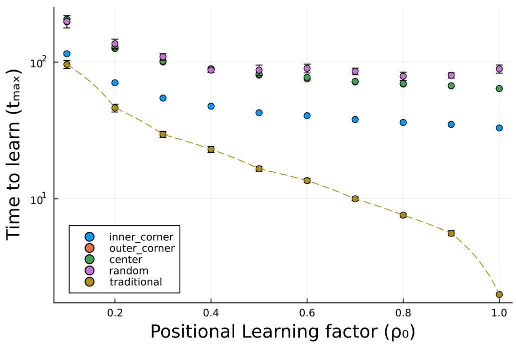
(b) $L = 48$



(c) $L = 64$



(d) $L = 96$



(e) $L = 128$

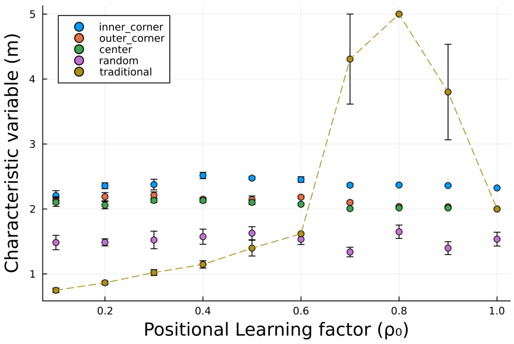
Figure 2.6: Time to learn t_{max} as a function of positional learning factor ρ_0 for different classroom sizes $L \in \{32, 48, 64, 96, 128\}$. Lower time to learn t_{max} indicate better performance.

2.3.2 Class learning rate m vs positional learning factor ρ_0

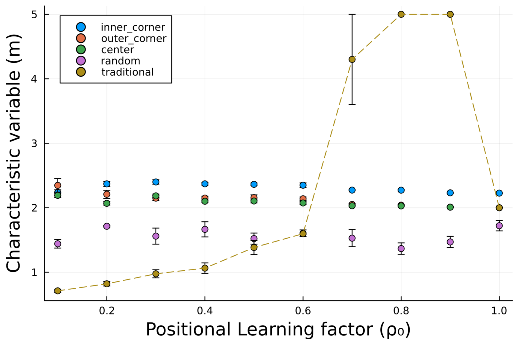
The data in Figure 2.7 shows that the class learning rate m does not generally change for different positional learning factors ρ_0 when comparing against similar SAs for PI. In traditional instruction, however, there is an increase in class learning rate m with increasing ρ_0 for all classroom sizes. Similar to the findings in Section 2.3.1, the inner corner SA has the highest learning rate m for all classroom sizes. The outer corner and center SA's performed similarly, while the random SA has the lowest class learning rate m. We still find that the traditional model performs situationally better than even the inner corner PI model.

The sudden decrease in class learning rate m for the traditional model at $\rho_0 = 1.0$ is due to the improper truncation and fitting of the data. In cases of traditional instruction where $\rho_0 = 1.0$, all the students will transition from unlearned to learned in one time step, so fitting the power law to the first 25% of the data would yield a class learning rate m that does not represent the dynamics of the simulations well.

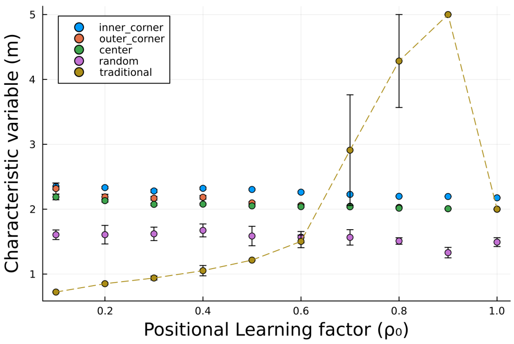
The results in this section show that traditional instruction performs better than PI in terms of class learning rate m for all classroom sizes at low ρ_0. This is in contrast to the results in Section 2.3.1 where PI performed better only in small classrooms with low positional learning factor ρ_0. This discrepancy is likely due to the different metrics used to evaluate the performance of the models. The class learning rate m is a measure of how quickly the students learn, while the time to learn t_{max} is a measure of how long it takes for all the students to learn. In cases where the classroom is large and positional learning factor ρ_0 is small, using traditional instruction enables the teacher to teach all the students in a shorter amount of time despite PI being able to teach the students more effectively.



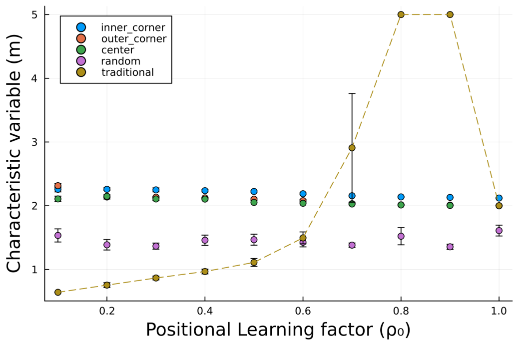
(a) $L = 32$



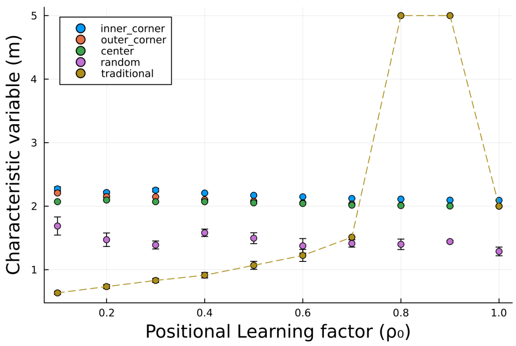
(b) $L = 48$



(c) $L = 64$



(d) $L = 96$



(e) $L = 128$

Figure 2.7: Class learning rate m as a function of positional learning factor ρ_0 for different classroom sizes $L \in \{32, 48, 64, 96, 128\}$. Higher class learning rate m values indicate better performance.

2.3.3 Time to learn t_{max} vs class size N

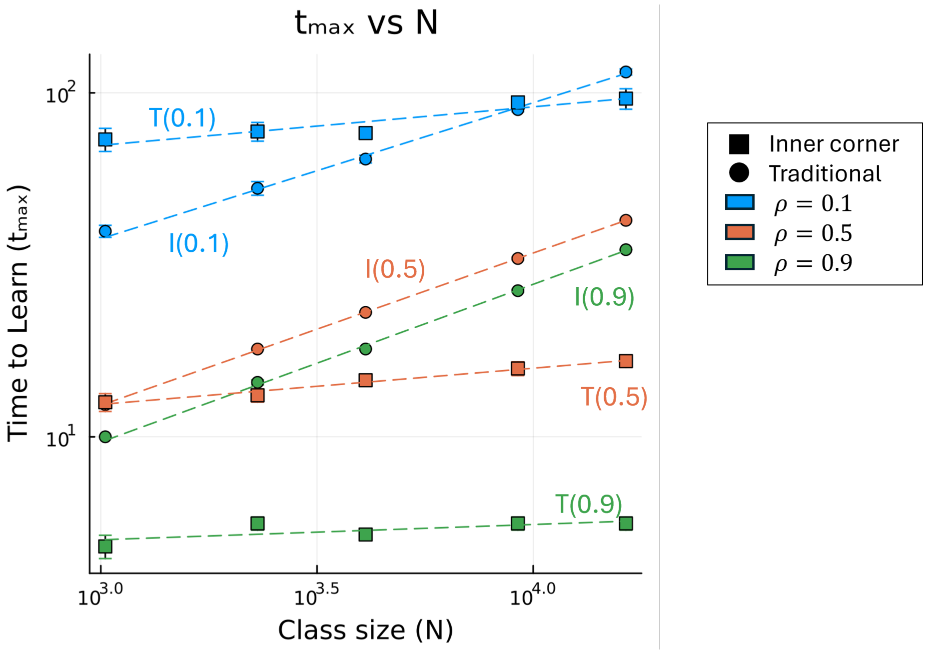


Figure 2.8: Time to learn t_{max} as a function of class size N for PI and traditional models. Circles and squares denote data points for PI (inner corner SA) and traditional instruction, respectively. The dash lines are the corresponding power law fit of the form $t_{max} = a \cdot N^b$ using $\rho \in \{0.1, 0.5, 0.9\}$. The fitted parameters are shown in the table with an average of $\langle b \rangle = 0.4532 \pm 0.019$ for the inner corner and $\langle b \rangle = 0.087 \pm 0.021$ for the traditional model. Lower time to learn t_{max} indicate better performance.

	Inner corner		**Traditional**	
	a	b	a	b
$\lambda = 0.1$	2.4515	0.3955	32.4884	0.1119
$\lambda = 0.5$	0.5795	0.4428	6.0079	0.1052
$\lambda = 0.9$	0.4055	0.4589	3.6914	0.0445

Table 2.1: Power law fit parameters a and b for t_{max} vs N, where $t_{max} = a \cdot N^b$

When we analyze the dependence of the time to learn t_{max} on the class size N, we find that the trend for both models follow the power law $t_{max} = a \cdot N^b$. In Figure 2.8 we observe transition points where the traditional instruction becomes more efficient than PI. These points happen at progressively low class sizes as the value of ρ_0 increases. We also find two major groups of power laws based on their b values, as summarized in Table 2.1. The first group has an average b value of $\langle b \rangle = 0.4532 \pm 0.019$ for the inner corner SA, while the second group has an average b

value of $\langle b \rangle = 0.087 \pm 0.021$ for the traditional instruction. The inner corner SA has higher b values than traditional instruction, indicating that traditional instruction is less affected by class size N and so is more scalable than PI.

2.4 Discussion/conclusions?

In this chapter, we have shown that between different seating arrangements for PI, the inner corner SA is the most efficient in terms of time to learn t_{max} and class learning rate m. The outer corner and center SAs performed similarly worse than the inner corner SA, while the random SA performed the worst. This is different from what existing literature has shown, where the outer corner SA performed the best [5]. This is likely because of the simplifications made in our model. Our model does not incorporate factors such as the similarity effect mentioned in previous studies [5, 6]. This effect is the phenomenon wherein students of similar aptitude levels tend to learn from each other better when seated together regardless their actual aptitude. Our model also does not consider anisotropic positional learning factors $\rho_{i,j}$, where the probability of learning from a neighbor varies with the neighbors' relative positions. Besides being binary, which introduces granularity, our model also does not consider that not all students are equally receptive to peer instruction or have the same learning rate. Introducing these factors, some of which we will explore in the next chapters, may make our model reflect reality better and provide us with better understanding of the learning dynamics in the classroom.

The nature of our model lends itself to be heavily influenced by geometric factors. The most impactful factor in this case is then the distance d_{max} of any student to a high aptitude or learned student as shown in figure 2.9. The inner corner SA minimizes d_{max}, which is why it is the most efficient. The outer corner and center SAs have equal d_{max} higher than the inner corner SA, which explains why they performed very similarly albeit worse than the inner corner SA.

Our analysis on the dependence of t_{max} on N gives us an explanation on why the traditional model performed better than the PI model in most cases. With lower b-values, the traditional model is less affected by class size than the PI models are. Because of this, the traditional model performed better in cases with larger classrooms.

With regards to the performance of PI methods with traditional teaching meth-

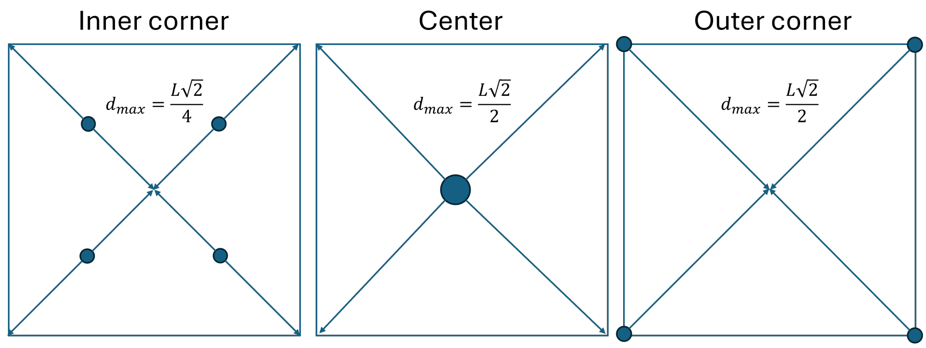


Figure 2.9: Distance d_{max} of any student to a high aptitude or learned student for different seating arrangements.

ods, we found that classrooms with higher ρ_0 values performed better in traditional instruction, while classrooms with lower ρ_0 values performed better in PI. A previous similar study [5] found that classes with lower aptitude levels were the ones who benefited the most from the PI methods. Although we cannot directly conclude this from our model, we were able to show that students with a lower probability of learning or students who learn slower, can benefit as much from PI as in traditional set ups. *This reinforces other studies' findings [2] that show that PI methods can be effective even regardless of the students' actual aptitude levels.*

Chapter 3

Heterogeneous Learning Rates

To introduce learning rate heterogeneity in the classroom, we revisit equation 2.3.

$$P_{ij} = 1 - \prod_{\forall \delta i, \delta j} [1 - (\lambda_{ij})(\rho_{i+\delta i, j+\delta j})(s_{i+\delta i, j+\delta j})] \tag{2.3 revisted}$$

We can adjust the parameter λ_{ij} to introduce differences in each student's learning rate. We set a student's learning rate as $\lambda_{ij} = \lambda_0 \pm \delta\lambda$ where $\delta\lambda \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$ and $\lambda_0 = 0.5$. Each student has an equal chance of having a learning rate that is either faster ($\lambda_0 + \delta\lambda$) or slower ($\lambda_0 - \delta\lambda$) than the average learning rate λ_0.

$\lambda_{i,j} \in [0, 1]$ is the learning rate of student $c_{i,j}$ with values $\lambda_{i,j} = \{\lambda_0 \pm \delta\lambda\}$ where $\lambda_0 = 0.5$ and $\delta\lambda = \{0.1, 0.2, 0.3, 0.4\}$,

3.1 Effects on classroom evolution

3.1.1 Peer Instruction (PI)

When introducing heterogeneity to PI set ups, we see that the general trend of the classroom evolution is still the same as the homogenous case. As shown in the figures in this section, we notice values of ρ_0 is still the most important factor in determining class performance. However, irregularities in the shapes of the "wave of learning" become more pronounced for lower values of ρ_0 and high values of $\delta\lambda$ as shown in Figure 3.1. These irregularities are also more pronounced at the start of the classroom and fades over time. For high values of $\delta\lambda$, low values of ρ_0 leads to the irregularity in the "wave of learning" to persist longer than those with higher values of ρ_0. The irregularities in the shape of the "wave of learning" and the deviation from the power law are less pronounced for higher values of ρ_0, regardless of the value of $\delta\lambda$, as shown in Figures 3.2 and 3.3. In these cases, the class learning rate is slower than the power law fit at the start of the simulation.

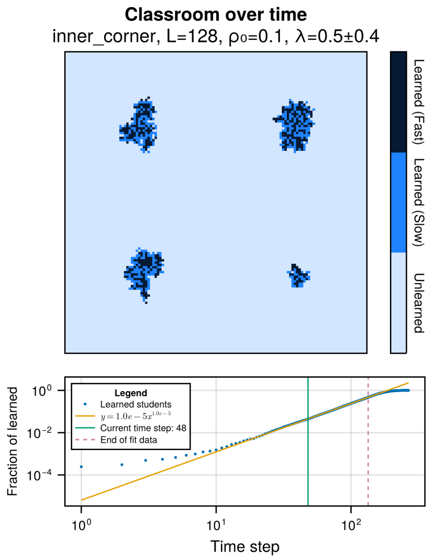
3.1.2 Traditional Instruction

For traditional instructon, even though the value of positional learning coefficient ρ_0 is still an important factor in determininng class performance, the effect of $\delta\lambda$ is more pronounced compared to PI. The trend we see in traditional instruction that is absent or less evident in PI, is that majority of the students that learn earlier in the simulations are fast learning students students (Figure 3.4(a)). After this period, most of the simulation time is spent waiting for the slow learning students to learn. This behavior is more pronounced for higher values of $\delta\lambda$ and lower values of ρ_0 as shown in Figure 3.4. This could explain the deviation from the power law fit at earlier times t for the traditional model considering that the deviation shows that the class learning rate is higher than the fitted power law.

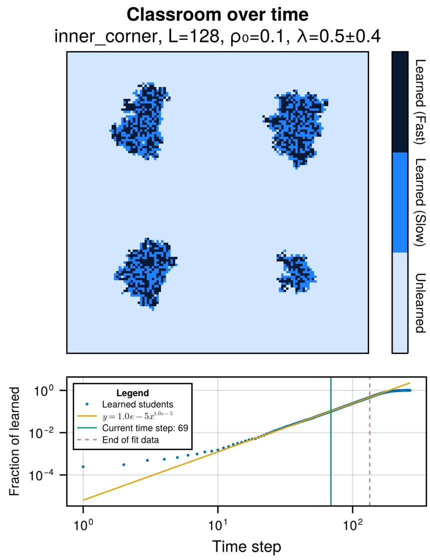
3.1.3 Comparing temporal learning dynamics of different classroom configurations

When varying the class size N, as in Figure 3.5(a), we see that in traditional instruction, the dynamics remains generally unchanged, with only the time to learn t_{max} increasing as the class size N increases. For PI, an increase in class size N adds a time delay to when the learning starts to speed up. Despite the added time delay, the general shape of the learning curve remains the same. An increase in class size N also does not affect the time to learn t_{max} for PI as much as it does for the traditional model.

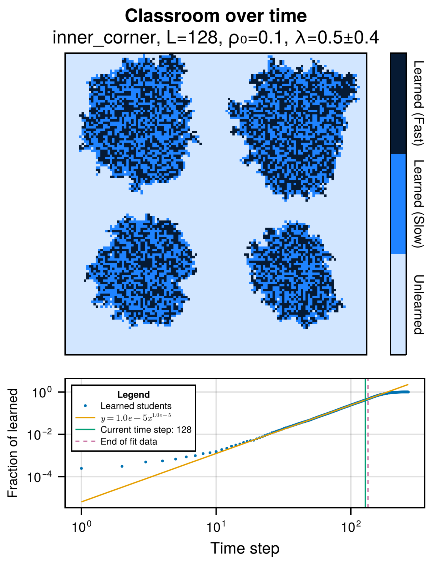
When varying the positional learning factor ρ_0, we see that it varies the shape of the learning curve for traditional instruction, especially for lower values of ρ_0 as shown in Figure 3.5(b). For traditional instruction, the value of ρ_0 also greatly changes the initial number of students that learn in the second time step. The same figure shows that for PI, the value of ρ_0 has a similar effect to class size N, where it only adds a time delay to when the learning starts to speed up. Notably, the effect of ρ_0 is not linear for either instruction models. We can see in the same figure that those with extremely low values of ρ, like that of $\rho_0 = 0.1$ perform much worse with the learning curves being further from $\rho_0 = 0.5$ than $\rho_0 = 0.9$.



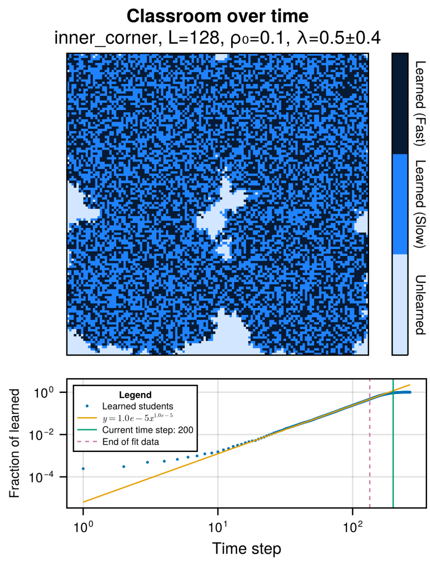
(a) $t = 48$



(b) $t = 69$

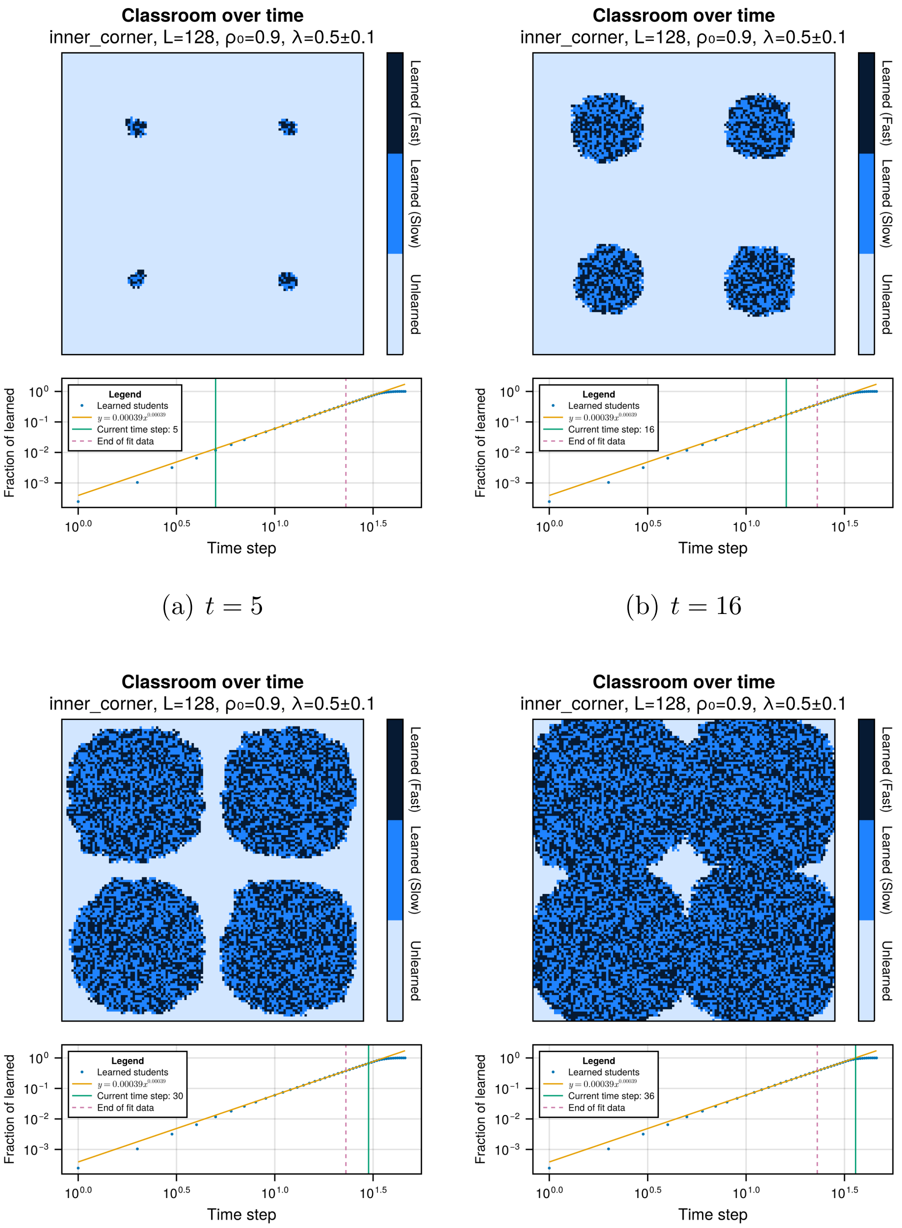


(c) $t = 128$



(d) $t = 200$

Figure 3.1: Sample classroom evolutions for PI with the inner corner SA with $L = 128$ at different times t for positional learning coefficient $\rho_0 = 0.1$, $\delta\lambda = 0.4$. Dark blue squares represent learned students with learning rate $\lambda = \lambda_0 + \delta\lambda$, blue squares represent learned students with learning rate $\lambda = \lambda_0 - \delta\lambda$, and light blue squares represent unlearned students. The accompanying graph shows the fraction of learned students as a function time step. The blue dots represent data points. The yellow line shows the power law fit. The pink dashed vertical line shows where we truncate the data for fitting the power law. The green vertical line shows the current time step in the simulation.



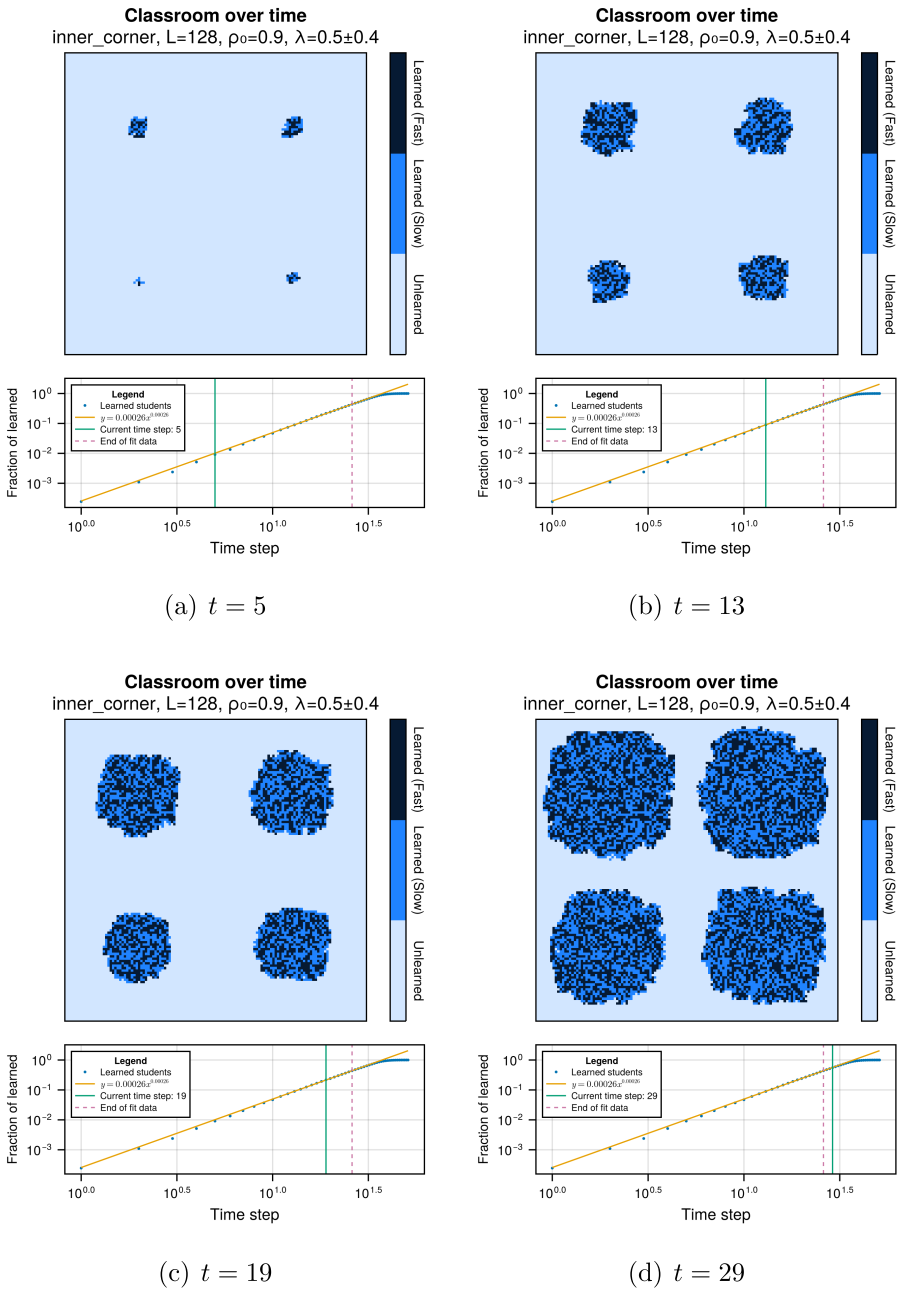
(a) $t = 5$

(b) $t = 16$

(c) $t = 30$

(d) $t = 36$

Figure 3.2: Sample classroom evolutions for PI with the inner corner SA with $L = 128$ at different times t for positional learning coefficient $\rho_0 = 0.9$, $\delta\lambda = 0.1$. Dark blue squares represent learned students with learning rate $\lambda = \lambda_0 + \delta\lambda$, blue squares represent learned students with learning rate $\lambda = \lambda_0 - \delta\lambda$, and light blue squares represent unlearned students. The accompanying graph shows the fraction of learned students as a function time step. The blue dots represent data points. The yellow line shows the power law fit. The pink dashed vertical line shows where we truncate the data for fitting the power law. The green vertical line shows the current time step in the simulation.



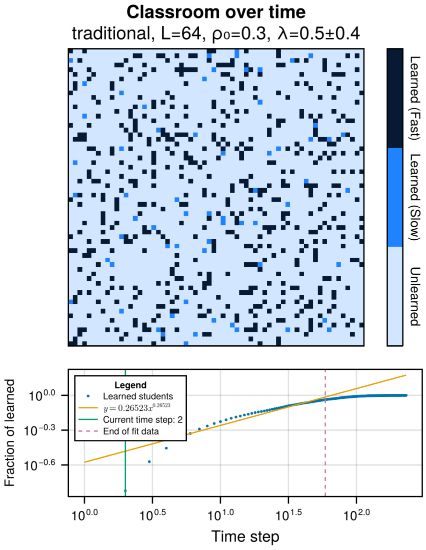
(a) $t = 5$

(b) $t = 13$

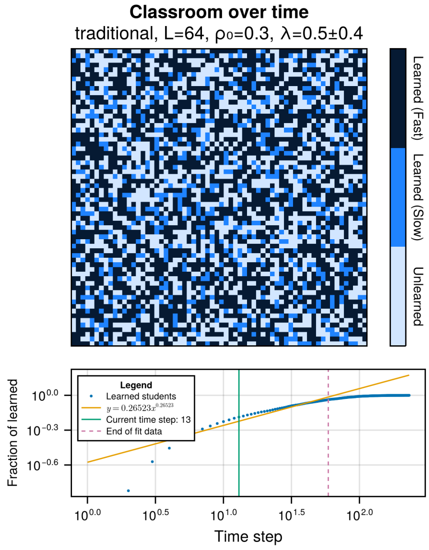
(c) $t = 19$

(d) $t = 29$

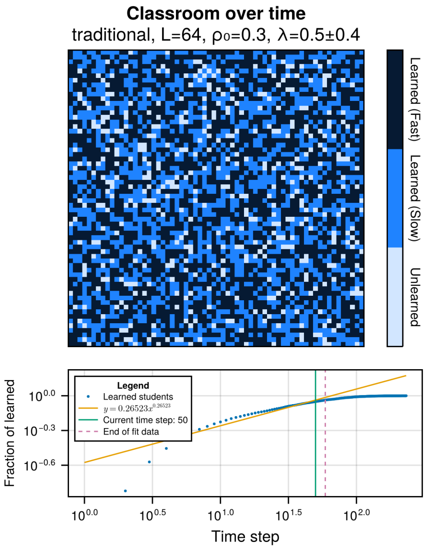
Figure 3.3: Sample classroom evolutions for PI with the inner corner SA with $L = 128$ at different times t for positional learning rate $\rho_0 = 0.9$, $\delta\lambda = 0.4$. Dark blue squares represent learned students with learning rate $\lambda = \lambda_0 + \delta\lambda$, blue squares represent learned students with learning rate $\lambda = \lambda_0 - \delta\lambda$, and light blue squares represent unlearned students. The accompanying graph shows the fraction of learned students as a function time step. The blue dots represent data points. The yellow line shows the power law fit. The pink dashed vertical line shows where we truncate the data for fitting the power law. The green vertical line shows the current time step in the simulation.



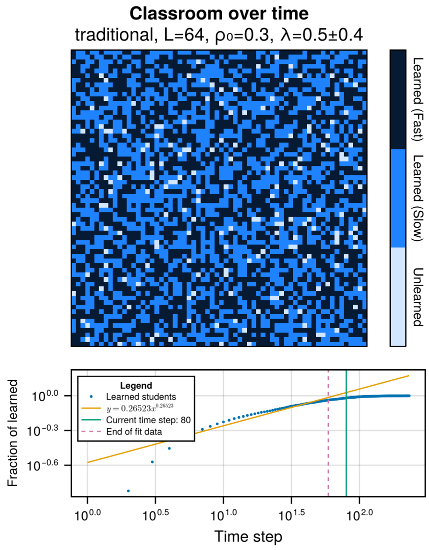
(a) $t = 2$



(b) $t = 13$



(c) $t = 50$



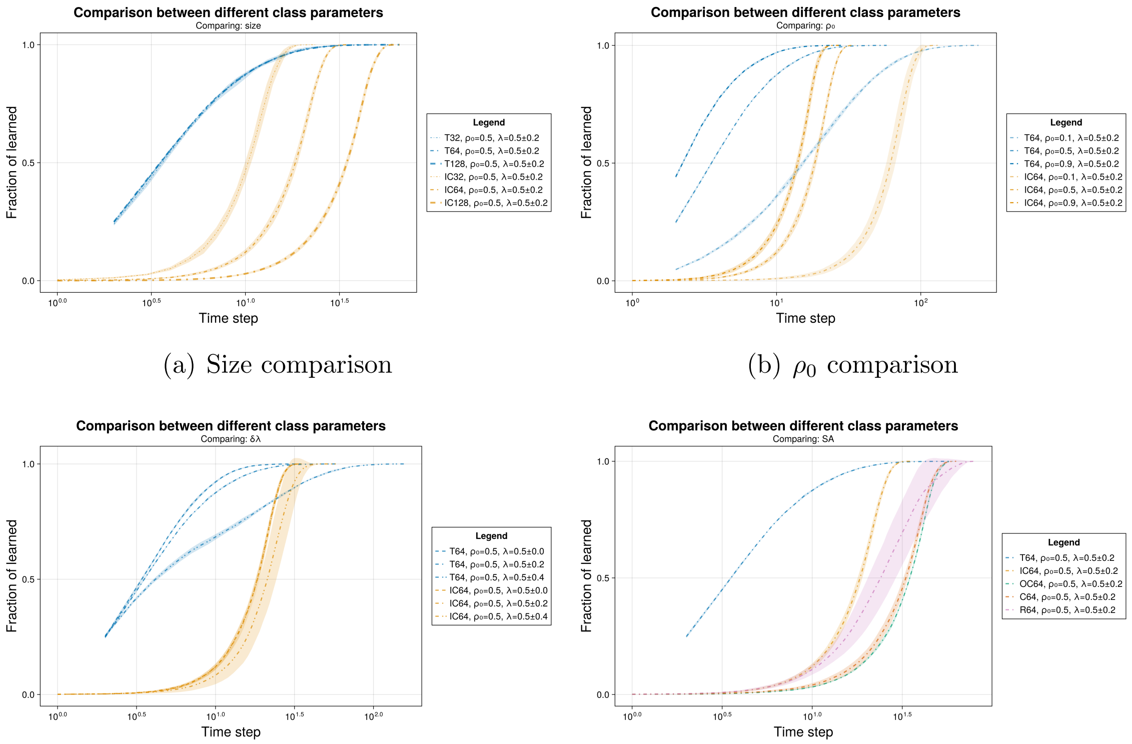
(d) $t = 80$

Figure 3.4: Sample classroom evolutions for PI with the inner corner SA with $L = 128$ at different times t for positional learning coefficient $\rho_0 = 0.9$, $\delta\lambda = 0.4$. Dark blue squares represent learned students with learning rate $\lambda = \lambda_0 + \delta\lambda$, blue squares represent learned students with learning rate $\lambda = \lambda_0 - \delta\lambda$, and light blue squares represent unlearned students. The accompanying graph shows the fraction of learned students as a function time step. The blue dots represent data points. The yellow line shows the power law fit. The pink dashed vertical line shows where we truncate the data for fitting the power law. The green vertical line shows the current time step in the simulation.

Figure 3.5(c) shows that an increase in heterogeneity $\delta\lambda$ changes how fast the slope of the learning curve changes for the traditional model without changing the initial number of learned students. As for PI, an increase in heterogeneity $\delta\lambda$ only affects the learning curve at very high heterogeneity $\delta\lambda$. The effect of heterogeneity for PI is also similar to the effects of class size N and positional learning factor ρ_0 where it only adds a time delay, shifting the learning curve to the right.

As shown in Figure 3.5(d), when comparing between the different SA's for PI and traditional instruction, at least for $L = 64, \rho_0 = 0.5, \lambda = 0.5 \pm 0.2$, traditional instruction generally performs better than PI, especially in the early time steps. Among the different SA's for PI, the inner corner SA still performs the best while the center and outer corner SAs perform similarly, the same conclusion we got from the homogenous case. Contrary to the results of the homoegnous classes, the random SA did not perform the worst, it performed better than the center and outer corner SA's.

One observation that is consistent with all the comparisons shown in Figure 3.5 is that even though using traditional instruction has more students learning in the early time steps, it also spend more time waiting for the last few students to learn. In contrast to this, PI takes some time before the learning starts to speed up, but once majority of the students have learned, it does not take long until all the students to learn. This makes PI have a shorter time to learn t_{max}, despite being outperformed by the traditional model in early time steps. This phenomenon is further investigated in the next sections where we look closer at the different factors that can affect which set up is better for different classroom parameters.



(a) Size comparison

(b) ρ_0 comparison

(c) $\delta\lambda$ comparison

(d) SA comparison

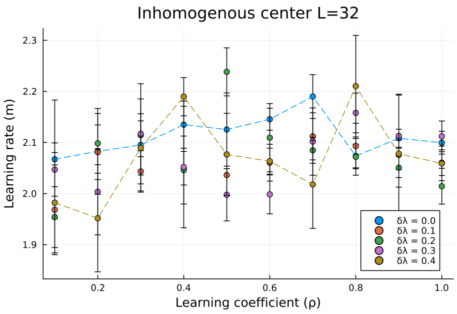
Figure 3.5: Comparison of time to learn t_{max} and fraction of learned students for different representative classroom configurations. Each SA corresponds to a different color, blue for traditional, yellow for inner corner, green for outer corner, orange for center, and pink for random. Different values ρ_0 corresponds to varying alpha or transparency, where lower ρ_0 values are more transparent. Different values of $\delta\lambda$ corresponds to different line styles, where $\delta\lambda = 0.0$ are represented by dashed lines, $\delta\lambda = 0.2$ are represented by lines with alternating dots and dashes, $\delta\lambda = 0.4$ are represented by two dots followed by a dash. Different classroom sizes L corresponds to different line widths, where bigger classroom sizes correspond to thicker lines. The bands around each line show the standard deviation of the data over 5 trials. Higher fraction of learned indicates better performance.

3.2 Class learning rate m vs positional learning factor ρ_0

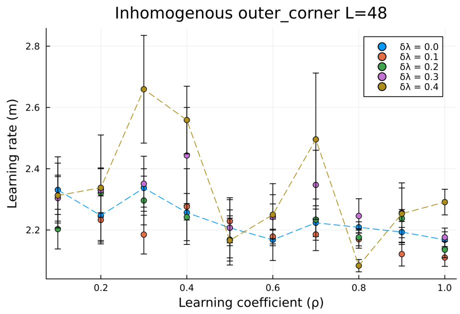
Figure 3.6 shows that class learning rate becomes inconsistent with the positional learning factor ρ_0 in PI models when introducing heterogeneity. The inconsistency may be caused by sampling problem where the first 50% of the data may not be just capturing the initial dynamics of the classroom, which should be the only basis of the class learning rate m. With heterogeneity, PI models no longer show trends in learning rate m as a function of positional learning factor ρ_0.

The traditional instruction model shows a similar trend in learning rate m as with time to learn t_{max} where homogenous classrooms perform better then heterogenous classrooms. This might be explained by the spatiotemporal dynamics of traditional learning models where the fast learning students learn first and the slow learning students learn last, as discussed in Section 3.1 and shown in Figure 3.4.

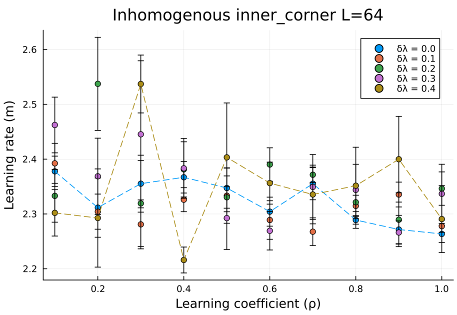
Moving forward, we will only focus on the time to learn t_{max} as a metric for performance.



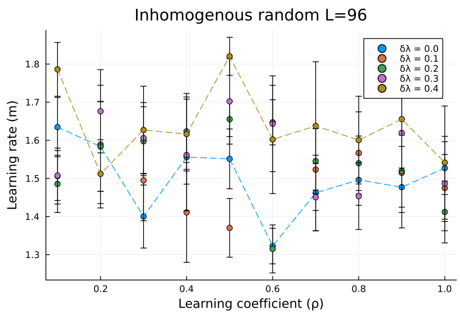
(a) Center $L = 32$



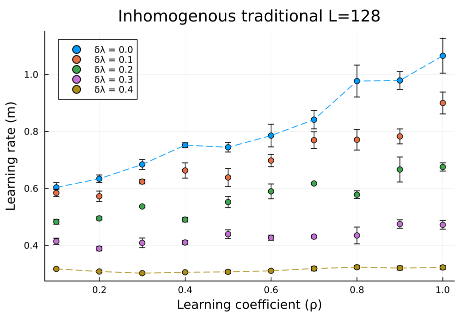
(b) Outer corner $L = 48$



(c) Inner corner $L = 64$



(d) Random $L = 96$



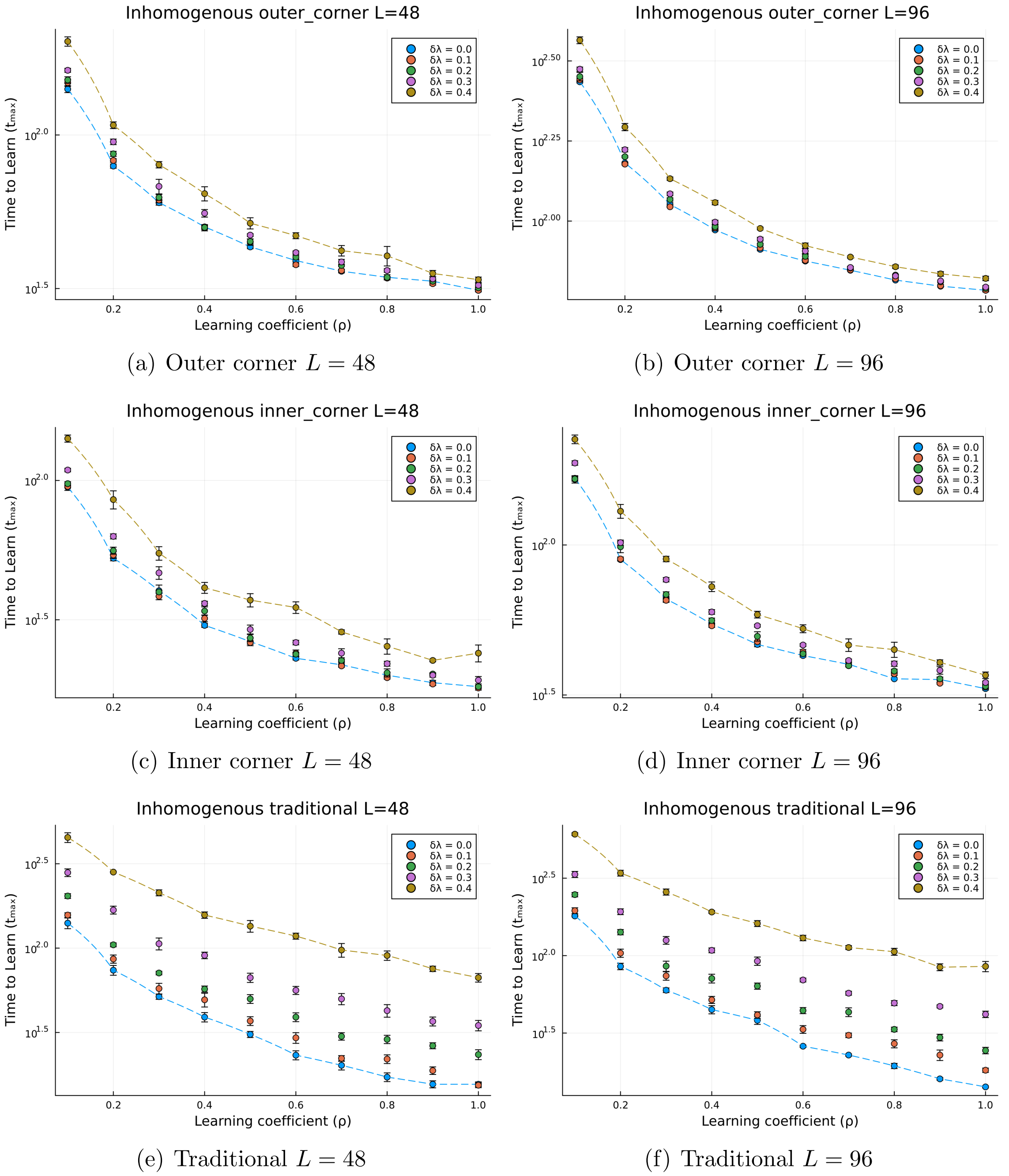
(e) Traditional $L = 128$

Figure 3.6: Representative plots for class learning rate m as a function of positional learning coefficient ρ_0. Each subplot represent a different SA and classroom size L. In each plot, the circles represent the data points, color represents a different value of $\delta\lambda \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$, dashed lines connect the data points for $\delta\lambda \in \{0.0, 0.4\}$. Higher class learning rate m values indicate better performance.

3.3 Time to learn t_{max} vs positional learning factor ρ_0

Figure 3.7 shows the time to learn t_{max} is directly affected by the level of heterogeneity $\delta\lambda$. This means that the heterogeneity $\delta\lambda$ has a significant effect on the time to learn t_{max}, with lower values of $\delta\lambda$ leading to lower time to learn t_{max} for all values of ρ_0. We also notice that the traditional model is more affected by classroom heterogeneity comapred to PI models.

When aggregating the different results for each class size L, as shown in Figure 3.8, it affirms our previous findings that PI models are better for smaller clases and lower values of ρ_0. This stays consistent when comparing the performance of a similar classroom size L and heterogeneity $\delta\lambda$ using the traditional model. However, the comparison is not as clear when comparing between different levels of heterogeneity $\delta\lambda$. When $L \geq 64$, the PI models can perform better than the traditional model depending on the heterogeneity $\delta\lambda$, even with the same positional learning factor ρ_0.



(a) Outer corner $L = 48$

(b) Outer corner $L = 96$

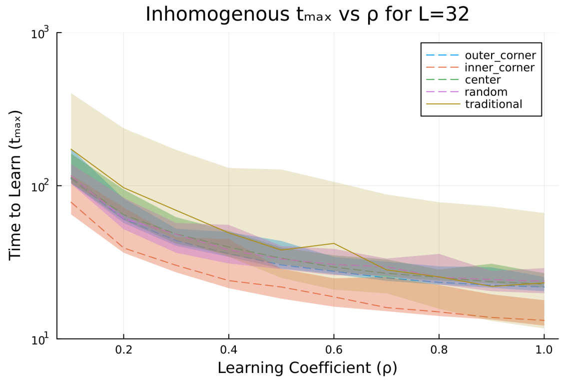
(c) Inner corner $L = 48$

(d) Inner corner $L = 96$

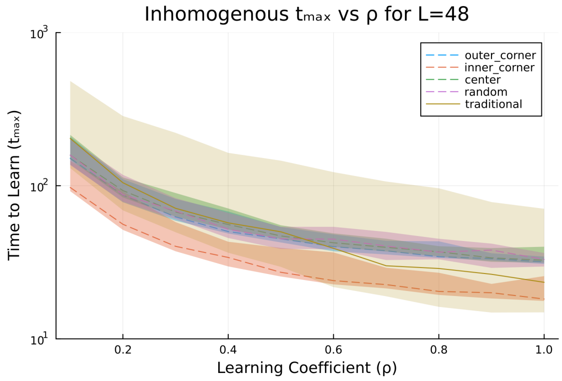
(e) Traditional $L = 48$

(f) Traditional $L = 96$

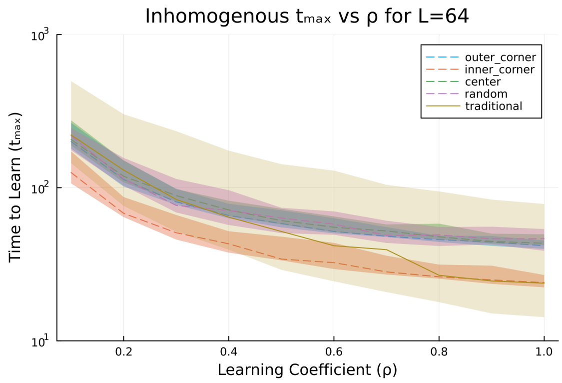
Figure 3.7: Time to learn t_{max} as a function of positional learning factor ρ_0 for different representative classroom configurations and sizes with varying heterogeneity $\delta\lambda \in \{0, 0.1, 0.2, 0.3, 0.4\}$. Lower time to learn t_{max} indicates better performance.



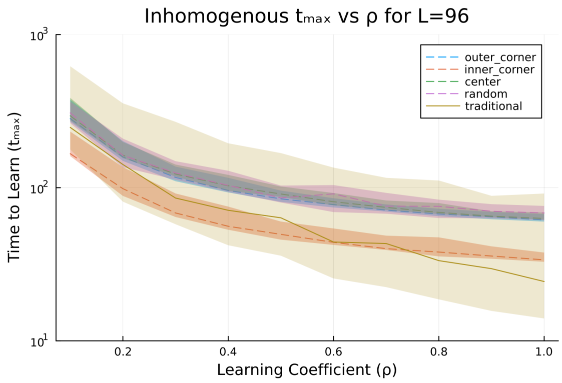
(a) $L = 32$



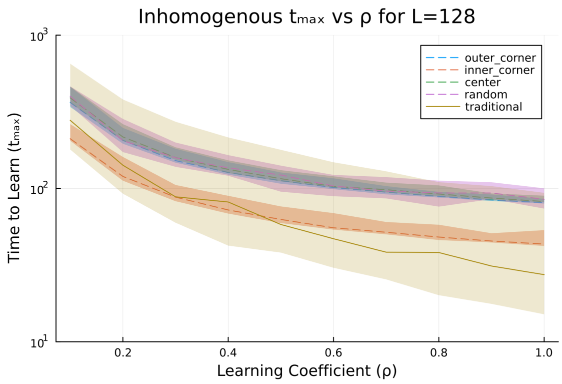
(b) $L = 48$



(c) $L = 64$



(d) $L = 96$



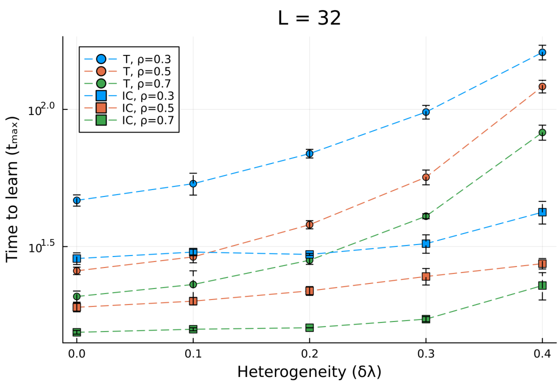
(e) $L = 128$

Figure 3.8: Time to learn t_{max} as a function of positional learning factor ρ_0 for the heteregenous model of all seating arrangements and classroom sizes L. Each ribbon series represent the range of values for t_{max} for a given seating arrangments with heterogeneity $\delta\lambda = \{0.0, 0.1, 0.2, 0.3, 0.4\}$. Lower time to learn t_{max} indicates better performance.

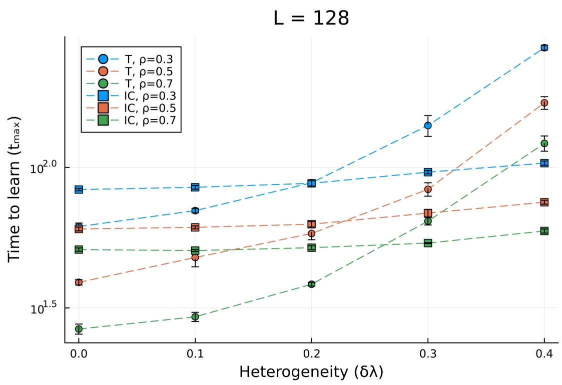
3.4 Time to learn t_{max} vs heterogeneity $\delta\lambda$

Figure 3.9 shows that as heterogeneity $\delta\lambda$ increases, PI methods perform better than traditional methods even when PI methods have lower positional learning factors ρ_0. For our represntative ρ_0 values, PI performs better than traditional methods regardless of the ρ_0 value, even in a large classroom $L = 128$. However, in larger classrooms, the advantage in performance of PI over traditional methods is less pronounced. Furthermore, as class size increases, traditional methods tend to stay advantageous than PI methods for increasing values of heterge $\delta\lambda$.

For example, at $L = 32$ shown in figure 3.9(a), PI methods are better than traditional methods for all values of $\delta\lambda$ when comparing between equal ρ_0 values. However, at $L = 128$ shown in figure 3.9(b), the traditional model performs better than the PI model for $0 \leq \delta\lambda \leq 0.2$ when comparing between equal ρ_0 values.



(a) $L = 32$

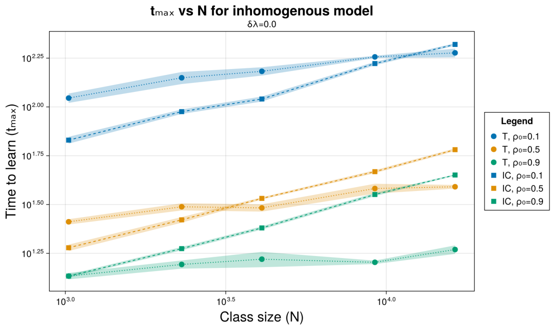


(b) $L = 128$

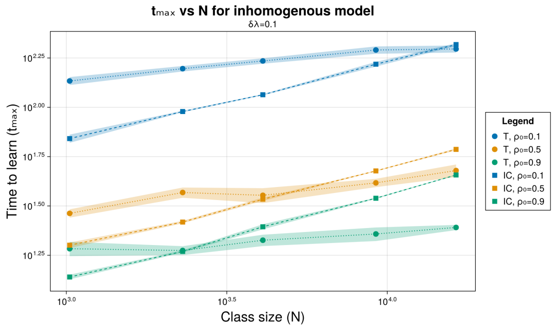
Figure 3.9: Time to learn t_{max} as a function of heterogeneity $\delta\lambda$ for the heteregenous models of the PI (inner corner SA) and traditional models with varying positional learning factor $\rho_0 \in \{0.3, 0.5, 0.7\}$ and classroom sizes $L \in \{32, 128\}$. Each color represents a diffent value of ρ_0, while the circle and square symbols represent the traditional and PI models respectively. Lower time to learn t_{max} indicates better performance.

3.5 Time to learn t_{max} vs class size N

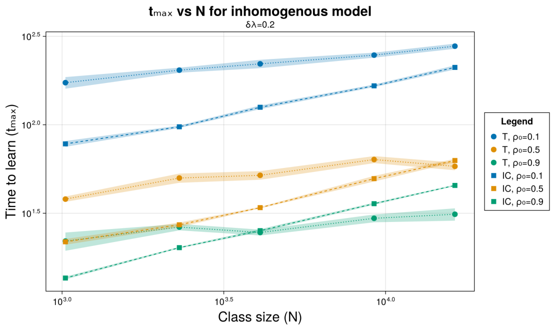
When investigating the size dependence of time to learn t_{max} shown in Figure 3.10, we see a similar trend to what we found in section 2.3.3 where PI models are better for smaller classes and traditional models are better for larger classes. Heterogeneity $\delta\lambda$ generally has a negative effect on the time to learn t_{max} for both PI and traditional models. However, as shown in sections 3.3 and 3.4, the traditional model is more affected by heterogeneity $\delta\lambda$ compared to PI models. Because of the sensitivity of traditional instruction to heterogeneity $\delta\lambda$, at low values of ρ_0, even for large classes, PI models can perform better than traditional models for higher values of heterogeneity $\delta\lambda$. This is shown in Figures 3.10(d) and 3.10(e) where the PI model performs better than the traditional model for all class sizes (N) and the same value of ρ_0.



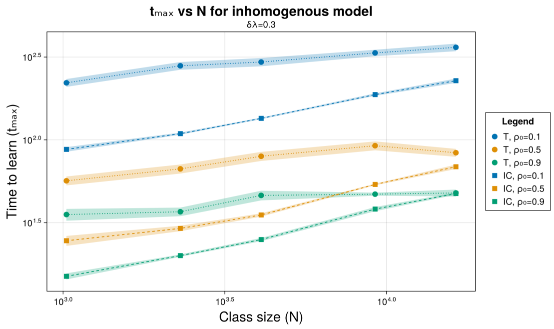
(a) $\delta\lambda = 0.0$



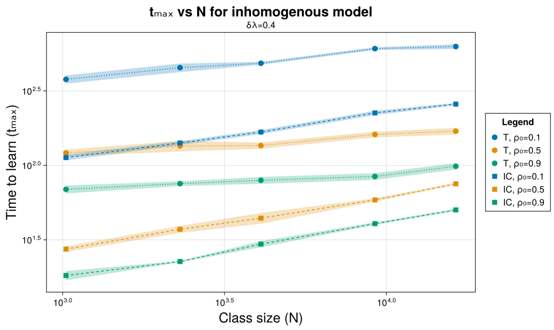
(b) $\delta\lambda = 0.1$



(c) $\delta\lambda = 0.2$



(d) $\delta\lambda = 0.3$



(e) $\delta\lambda = 0.4$

Figure 3.10: Time to learn t_{max} as a function of class size N for the heteregenous models of the PI (inner corner SA) and traditional models with varying positional learning factor $\rho_0 \in \{0.1, 0.5, 0.9\}$ and heterogeneity $\delta\lambda \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$. Each color represents a diffent value of ρ_0, while the circle and square symbols represent the traditional and PI models respectively. Each band represent the standard deviation of time to learn t_{max} over 5 trials. Lower time to learn t_{max} indicates better performance.

3.6 Discussions/Conclusions?

To better model the real world, we introduced heterogeneity to the students' learning rate. Adding heterogeneity does not affect the general trend of our instruction models. For the case of a heteregenous system, we still find traditional instruction fares better than PI for larger classes with high positional learning factor rho_0.

Besides not affecting the general trend, we also found out that traditional instruction is more sensitive to heterogeneity. As discussed in section 3.1, the difference in sensitivity is because of the dynamics that were seen in the classes' evolution. This is related to the spatiotemporal dynamics of traditional learning models where the fast learning students learn first and the slow learning students learn last. The time to learn t_{max} for traditional instruction is then dependent on waiting for the slow students.

In contrast to the spatiotemporal dynamics of traditional instruction, the advantage of PI is that it is not as heavily dependent on the students themselves as traditional instruction. PI is more dependent on the geomety of the classroom and the number of learning sources available to the students. For the case of PI, the "wavefront of learning" (see: section 3.1) provides students with more than one source of learning, unlike in the traditional model. This phenomenon offsets the increase in the time to learn t_{max} that is caused by waiting for the slow learning students. PI models therefore have a shorter time to learn t_{max} despite being outperformed by the traditional model in earlier time steps.

Given the foregoing, initially performing traditional instruction and then proceeding with PI later would maximize each methods' strengths. The traditional instruction phase would yield high learning rates at the start of the simulation, by allowing the fast learning students to learn first. The learned students would then be able to help the slow learning students learn in the PI phase, shortening the time to learn t_{max}.

Chapter 4

Mixed Model Instuction

In this chapter, a sample for making equations and sub-equations are demonstrated.

4.1 Equations and sub-equations

In the following, a set of equations is shown.

$$\vec{\nabla} \cdot \vec{D} = \rho \tag{4.1a}$$

$$\vec{\nabla} \cdot \vec{B} = 0 \tag{4.1b}$$

$$\vec{\nabla} \times \vec{E} = -\partial_t \vec{B} \tag{4.1c}$$

The last one being

$$\vec{\nabla} \times \vec{H} = \vec{J} + \partial_t \vec{D} \tag{4.1d}$$

Note that text can still be placed between sub-equations within the `subequations` environment.

When using a solitary equations, you may use the usual equation syntax in LaTeX.

$$E = mc^2 \tag{4.2}$$

Chapter 5

Conclusions

A short sample thesis/dissertation is presented. Although not complete, it will be useful for newbies in LaTeX. Any questions? email me at the following address: `johnrob.bantang@gmail.com`.

Appendix A

Sample appendix

This gives an example of an appendix chapter. Note that this file has been included **after** the line `\appendix` in `main.tex`.

A.1 Derivation of Equation 2.3

For any event e, the desired outcome occurs with probability p or not with probability q where

$$p + q = 1. \tag{A.1a}$$

For n events, each being event e:

$$\prod_{\forall e}(p_e + q_e) = 1. \tag{A.1b}$$

Expanding equation A.1b, we get

$$\prod_{\forall e} p_e + \ldots + \prod_{\forall e} q_e = 1. \tag{A.1c}$$

where the sum of the first $n - 1$ terms is the probability of the desired outcome occurring at least once over n events and the last term is the of the probability of the desired outcome not occurring at all. Thus, we can rewrite the probability of the desired outcome occurring at least once as

$$P = 1 - \prod_{\forall e} q_e. \tag{A.1d}$$

substituting equation A.1a into equation A.1d, we get

$$P = 1 - \prod_{\forall e}(1 - p_e). \tag{A.1e}$$

A.2 Equations in appendix

You don't need to worry about equations within the appendix since LaTeXautomatically formats the equation numbers for you. For example,

$$c^2 = a^2 + b^2 \tag{A.2}$$

becomes the Pythagorian theorem where c is the length of the longest side of any right triangle.

A.3 Codes as appendix

Include your codes when necessary to your thesis/dissertation. To do this, you may use `verbatim` environment as follows. **WARNING:** All verbatim and verbatiminput environments should always be treated as a separate paragraph. When included in a text paragraph, it sometimes happen to reduce the 1.5 spacing to the usual single-spaced text.

```
#include <iostream>
using std::cout;
using std::endl;

int main( void )
{
   cout << "Hello world!" << endl;
   return 0;
}
```

The `{\small }` bracketed region is used to lower the font size of the entire verbatim text. This will save you much space and give a more aesthetical look in your manuscript.

On the other hand, when very long codes are wished to be included automatically without the tedious cut and paste procedure, you may include them using the `\verbatiminput` command as follows. You may want to include a short description of the code of course.

```
//Johnrob Y. Bantang, Natinal Institute of Physics
//Created: 03 October 2002
// Makes new C files
// usage: newC filename
//Modifications:
```

```
// >> 21 Jan 2003, Johnrob
//   included the constant AUTHOR and AFFILIATION for portability

#include <stdlib.h>
#include <iostream.h>
#include <fstream.h>
#include <strstream.h>
#include <time.h>
#include <string.h>

const char *const AUTHOR= "Johnrob Y. Bantang";
const char *const AFFILIATION= "National Institute of Physics";
const char *const EXTENSION= ".cpp";

int main(int argc,char **argv){
if(argc!=2){
cout<<"usage: newC filename"<<endl;
exit(0);
}
char *fname= new char[strlen(argv[1])+5];
ostrstream out(fname,strlen(argv[1])+5);
out<<argv[1]<<EXTENSION;
time_t date=time(NULL);

ofstream file(fname,ios::nocreate,0);
//opens normal file that **already exists**;
if(file){
for(int i=1;i<5;i++)
cout<<"WARNING! file aready exists!"<<endl;
cout<<endl<<"you can type"<<endl<<endl;
cout<<"\t\"head "<<fname<<"\""<<endl;
cout<<"in command line to *view version*"<<endl<<endl;
cout<<"please enter 1 to OVERWRITE this file"<<endl;
cout<<"type anything to cancel"<<endl;
for(int i=1;i<5;i++)
cout<<"WARNING! file aready exists!"<<endl;
int n;
cin>>n;
if(n!=1){
cout<<"*no* file is created... exiting..."<<endl<<endl;
exit(0);
}
file.close();
file.open(fname);
if(!file)
cout<<"**cannot create new file!!**"<<endl;
cout<<"OLD FILE: "<<fname<<" *overwritten!*"<<endl;
}
if(!file){
```

```
file.open(fname);
if(!file){
cout<<"**cannot create new file!!**"<<endl;
exit(0);
}
cout<<endl<<"NEW FILE created: "<<fname<<endl<<endl;
}

cout<<"\tcreating contents for the new C++ file: "<<endl<<endl;
//creating headers...
file<<"//filename: \""<<fname<<"\""<<endl;
file<<"//"<<AUTHOR<<", "<<AFFILIATION<<endl;
file<<"//Created: "<<asctime( localtime(&date) );
//writes the time and date today; endl already in asctime();
file<<"//"<<endl;
file<<"//Comments:"<<endl;
file<<"// >>"<<endl;
file<<"//"<<endl;
file<<"//This file is generated using the \"newC generator\"..."<<endl;
file<<"//Modifications:"<<endl;
file<<"// >>"<<endl<<endl;
file<<"#include <iostream.h>"<<endl;
file<<"#include <math.h>"<<endl<<endl;
//starting the main body...
file<<"int main(int argc,char **argv){"<<endl;
file<<"//\tif(argc!= ...){"<<endl;
file<<"//\t\tcout<<\"usage: "<<argv[1]<<".exe ... \"<<endl;"<<endl;
file<<"//\t\texit(1);"<<endl;
file<<"//\t}"<<endl;
file<<"\t//write the main body here"<<endl;
file<<"return(0);"<<endl<<"}"<<endl;

delete fname;
  file.close();
  cout<<endl<<"\tCREATION SUCCESSFUL!"<<endl<<endl;
return 0;
}
```

This time, you may just include your recent codes by just copy-paste-ing the codes (as long they are clean!) into the directory `codes/` in the directory where this file is saved.

Bibliography

[1] M Arciaga, M Pastor, R Batac, J Bantang, and C Monterola. Experimental observation and an empirical model of enhanced heap stability resulting from the mixing of granular materials. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(07):P07040, 2009.

[2] Nathaniel Lasry, Eric Mazur, and Jessica Watkins. Peer instruction: From harvard to the two-year college. *American journal of Physics*, 76(11):1066–1069, 2008.

[3] Pierre-Yves Louis and Francesca R Nardi. *Probabilistic Cellular Automata: Theory, Applications and Future Perspectives*, volume 27. Springer, 2018.

[4] Hideo Nitta. A mathematical model of peer instruction and its applications. *Upgrading Physics Education to Meet the Needs of Society*, pages 87–98, 2019.

[5] RM Roxas, SL Carreon-Monterola, and C Monterola. Seating arrangement, group composition and competition-driven interaction: Effects on students' performance in physics. In *AIP Conference Proceedings*, volume 1263, pages 155–158. American Institute of Physics, 2010.

[6] Michelle K Smith, William B Wood, Wendy K Adams, Carl Wieman, Jennifer K Knight, Nancy Guild, and Tin Tin Su. Why peer discussion improves student performance on in-class concept questions. *Science*, 323(5910):122–124, 2009.