Memory Capacity in a System of Swelling Particles

David Keating
Advisor: Dr. Hilary Jacks

Department of Physics
California Polytechnic State University, San Luis Obispo

March 2022

Part I
Introduction

A system has the ability to store memory if one is able to write, retrieve, and erase information from it. Some systems are capable of storing multiple transient memories; with no noise in these systems, at long time, the memories degrade until one or two remain. The addition of noise to these systems can extend the retention of multiple memories, in some cases indefinitely [2, 3]. While this behavior was first observed in simulations of charge density waves, it has since appeared in other systems [1].

Past research has shown that sheared non-Brownian liquid suspensions of particles exhibit similar behavior with regards to multiple memory storage with noise [2, 3, 5]. In these systems, a memory of the driving shear that a system has been trained with can be written. Training consists of many cycles of driving at a given strain amplitude γ. After training, there is a sharp increase in irreversibility observed for shearing with a strain amplitude of $\gamma' > \gamma$ [2]. As a result, we can say that the system has formed a memory at γ.

In order to write multiple memories in a system, multiple amplitudes are used for the training in some predetermined pattern along with noise. Suppose we intended to train a system with two memories, γ_1 and γ_2 where $\gamma_1 > \gamma_2$. In order to train the system to have memories at both amplitudes we may repeat a pattern such as γ_1, γ_2, γ_2, γ_2, γ_2. The system is sheared to each amplitude successively, with noise applied between each shear, and then the pattern repeats. After sufficient repetitions of this pattern, multiple memories are detectable and persist at long time [3, 4].

While so far we've focused on sheared systems, past research by Keim, Paulsen, and Nagel indicates that multiple transient memory behavior is not sensitive to the deformation geometry of the system; for example, a system of swelling particles exhibits similar behavior [3]. This is the type of system used in this study. In this case, training the system is not done by shearing the suspension of particles to a given amplitude or pattern of amplitudes. Instead, the system is trained by swelling particles to a specified size or repeated pattern of sizes.

In these swelling systems, we investigate what system parameters influence memory capacity. Memory capacity can be defined as the total number of memories that can be written, stored, and read within one system at the same time. Many parameters about a given system, e.g. the amount of noise, the interaction strength, and the number of particles, may affect the memory capacity.

Part II
Methods

In this study a Python simulation is used. N particles are distributed in a static square box with side lengths l_{box} and periodic boundaries. Each particle is assigned an x, y coordinate from a uniformly distributed random value in the set [0, l_{box}).

Each cycle begins in the application of system-wide noise. In this system, the noise is applied to only a fraction of the total particles in the system. This fraction of particles is a tuneable parameter, ϵ_{noise}. At the beginning of each cycle, the simulation randomly selects an integer number of particles from all particles in the system. These particles then have their positions randomly set within the system, according to the same uniform distribution used in originally placing the particles. This noise can be understood as randomly picking up a percentage of particles in the system at the beginning of each cycle and dropping them at a new random location.

The area fraction, af, is the total area of the particles when swelled divided by the total area of the box. Given a system of N particles which swell to a radius of r in a box with side lengths l_{box}, the area fraction determines the swelling radius as:

$$af = \frac{N\pi r^2}{{l_{box}}^2} \tag{1}$$

In a given swelling cycle, all particles swell to a radius of r, where r and af are related by **Equation 1**. This is simulated by taking the coordinates of each particle and querying for neighboring particles, including those across the periodic boundary, separated by a distance [0, $2r$). These particles are seen as overlapping and therefore considered to be active. The total number of active particles in a given cycle is given by N_{act}. For any given swelling cycle the fraction of active particles f_{act} is given by:

$$f_{act} = \frac{N_{act}}{N} \tag{2}$$

When particles are tagged as active they are uniaxially repelled by a uniformly distributed random distance [0 d_r), where d_r is a tuneable parameter known as the "kick". In one cycle a particle will be repelled from all of its overlapping neighbors, and as a result the total path length of the particle in a given cycle is the sum of all

the individual repulsions from its active neighbors. Any particle whose displacement places it outside of the box boundaries is instead wrapped around to the opposite side via the periodic boundary condition.

While this is the total process for a given swelling cycle, we often make the distinction between a swelling cycle and a training cycle. A swelling cycle is as described above, where noise is applied followed by the repelling of particles tagged as active as determined by the area fraction. Note that noise and kick are both applied in every swelling cycle. A training cycle is defined as a set of swelling cycles corresponding to each area fraction in a sequence. For example, we may wish to train the system at multiple area fractions simultaneously. In such a case we may use a sequence such as af_1, af_2, af_2, where $af_1 > af_2$. In this example, one training cycle would consist of three individual swelling cycles in sequence.

With the mechanism for training a system established, the question of when training is considered complete arises. In a system with no noise this is simple to define: a noiseless system is fully trained when $N_{act} = 0$ for the largest value of af used in training. In a system with noise the question is more challenging. While for any given cycle it may be the case that $N_{act} = 0$, the application of noise means that the same may not hold true in a future cycle. Instead, a system with noise is considered trained when it reaches a statistical steady state. In this state, while N_{act} can be nonzero and may even fluctuate in successive cycles, over a number of cycles $< N_{act} >$ is constant. In this study we confirm that a system is fully trained by taking three samples of the system at sufficiently spaced training cycles to determine that no significant changes are still taking place.

Once a system has been trained it becomes necessary to have a mechanism by which the trained memories can be read. This is done by first creating a list of area fractions from 0 to 1. For the purposes of this analysis, 100 linearly spaced area fractions were used. Then N_{act} is determined for each area fraction in the list, though the particles are not repelled. For clarity a numerical derivative is then taken, as peaks are more easily identified than cusps in plots. As such, the plots shown can be seen as the rate of change of the tagged (active) particles as a function of area fraction N_{act}, or the "tag rate" for brevity.

Part III
Results

In this study, multiple parameters were investigated. This was accomplished by generating data for the system many times, varying one parameter at a time in order to understand how that parameter affects the system's memory capacity.

0.1 Writing algorithms

There are many methods which can be used to write multiple memories to the system. Three methods were tested in order to select the method that led to the most distinct peaks corresponding to the written memories. The first method was to run through a list of different area fractions repeatedly, such as $[af_1, af_2]$. The specified pattern was then looped c times, where c is the number of training cycles. This successfully wrote two memories, but the smaller of the two memories was significantly less prominent (Fig. 1A). Second, in the list of area fractions, the smaller memory was repeated, using a list like $[af_1, af_2, af_2]$ and the pattern was then looped c times. In cases of more than two memories each smaller value of af was repeated more times than the previous one. This gave better results, as the smaller memory was much more visible without the larger memory being significantly affected (Fig. 1B). A third attempted method was to write a smaller memory for half of the training cycles, then the larger memory for the remainder. This resulted in a memory being visible at the larger value, but the smaller memory fading significantly (Fig. 1C). Lastly, we attempted to write a larger memory completely, followed by the smaller memory. This resulted in only the smaller of the two memories being present when the system reached statistical steady-state, and so this method was deemed unsuccessful (Fig. 1D). After comparing these methods, the second method was selected for use for the rest of the study.

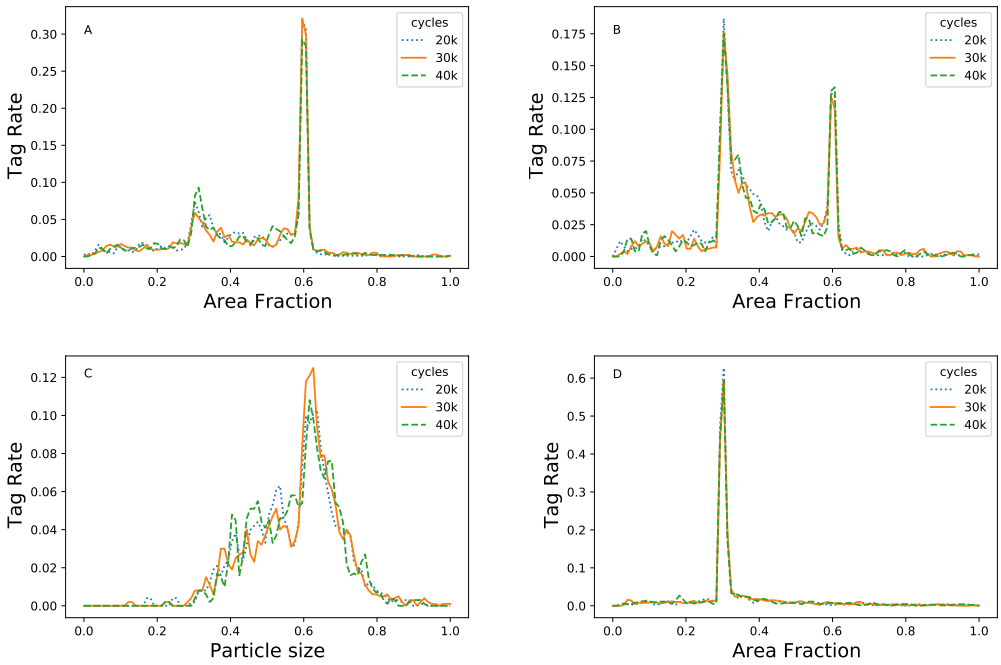


Figure 1: *Four different attempted writing algorithms for two memories, $af_1 = 0.6$ and $af_2 = 0.3$, in a system with noise. Each plotted line corresponds to the readout at a given number of training cycles. (1A) Alternating area fractions with smaller area fraction repeated, i.e. (af_1, af_2, af_2), repeat. (1B) Alternating area fractions with smaller area fraction not repeated, i.e. (af_1, af_2), repeat. (1C) Memories trained sequentially where af_1 was trained for 15,000 cycles, then af_2 was trained for the remainder. (1D) Memories trained sequentially where af_1 was trained for 10,000 cycles, then af_2 was trained for the remainder. The parameters used for all algorithms were as follows: l_{box}=40, N=1000, d_r=0.005, ϵ_{noise}=0.001, n=0.625*

0.2 Noise

Because noise is essential for the storage of multiple memories, we established a functional range of noise values. Three systems with identical parameters except for noise were trained with an area fraction of 0.5 for 30,000 cycles (Fig. 2). The system with no noise displays the memory written clearly, with a peak high enough that it was cut off for clarity of the other data. For this system, a minimum noise value of 0.0001 (1 particle per cycle) shows a clear memory but already begins to display evidence of the system's untrained state. For a larger noise value of 0.001 (10 particles per cycle) the particles' positions are randomly set frequently enough that nearly all traces of the memory are removed, leaving only the untrained state. Larger values of noise were not plotted because they show only the minor difference from the noise=0.001 plot of the small peak at af=0.5 disappearing, and so did not better illustrate the trend. The "optimal" value of noise is not independent, in that a higher particle density will have a lower optimal noise value. The exact nature of this relation is not investigated in this study.

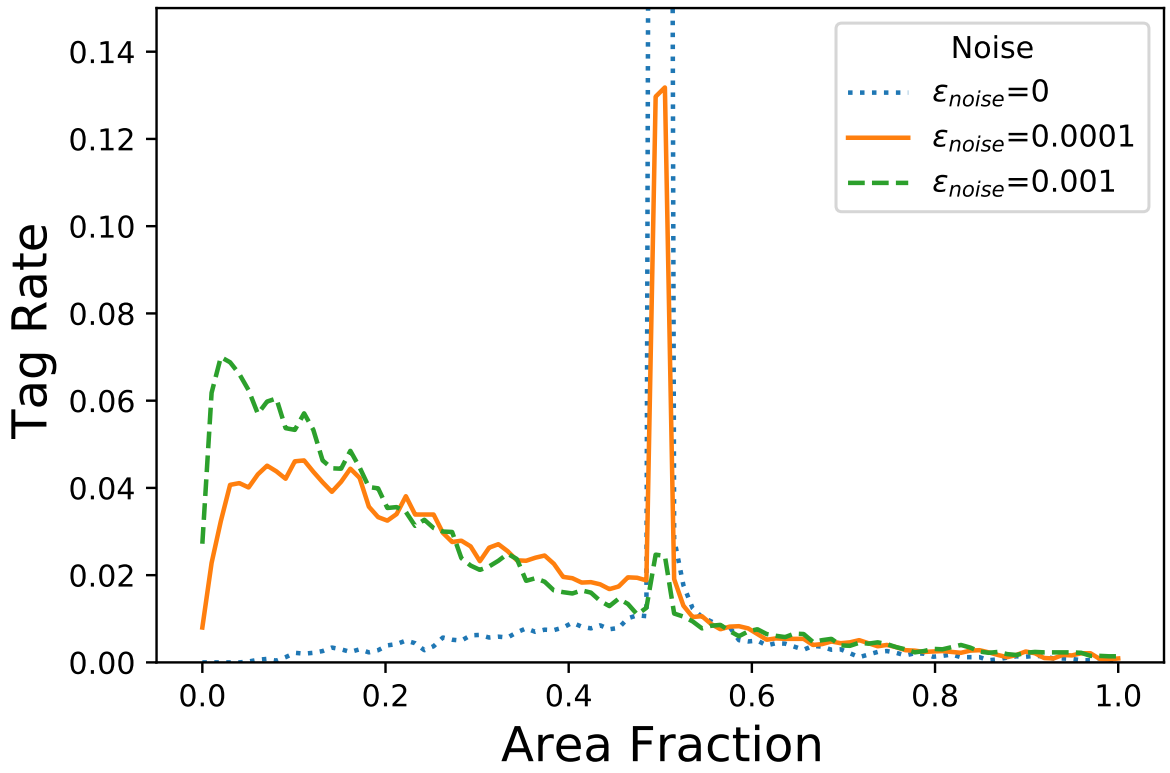


Figure 2: *Thresholds of behavior based on noise value. Three systems with identical parameters except for noise were trained with an area fraction of 0.5 for 30,000 cycles (Fig. 2). The system with no noise displays the memory written clearly, with a peak high enough that it was cut off for clarity of the other data. For this system, a minimum noise value of 0.0001 (1 particle per cycle) shows a clear memory but already begins to display evidence of the system's untrained state. For a larger noise value of 0.001 (10 particles per cycle) the particles' positions are randomly set frequently enough that nearly all traces of the memory are removed, leaving only the untrained state. Larger values of noise were not plotted because they show only the minor difference from the noise=0.001 plot of the small peak at af=0.5 disappearing, and so did not better illustrate the trend. The parameters that were the same between tests were as follows:* l_{box}*=12000, N=10000,* d_r*=0.005, n=6.94e-5.*

0.3 System Size

Table 1 and Fig. 3 show the effect of N, the number of particles in the system while keeping the particle density, n, constant. The number of particles in the system has a more significant impact on the ability to read memories from the system than it does the training dynamics of the system. Various area fractions were used to illustrate the clarity of each memory. Systems of about 100 particles or less show that an insufficient number of particles leads to largely unusable readings due to the tag rate not being smooth enough. This problem begins to be alleviated once N$\sim$1000 and is further alleviated as N is increased.

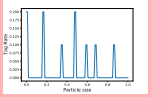
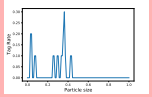
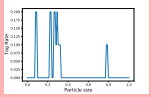
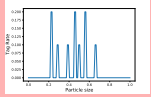
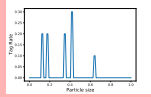
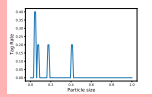
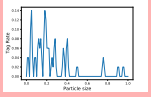
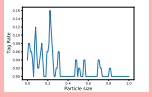
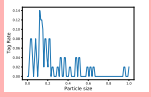
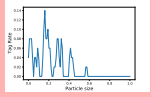
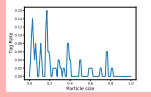
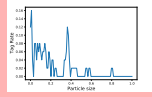
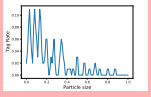
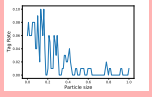
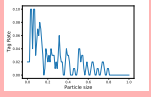
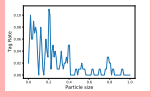
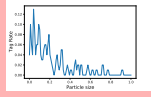
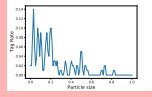
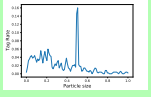
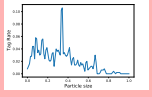
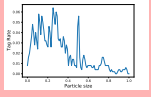
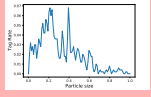
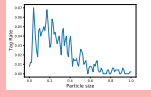
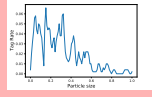
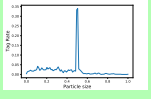
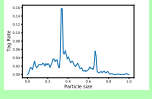
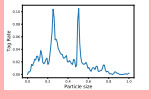
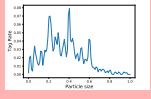
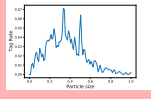
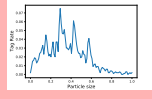
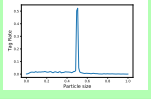
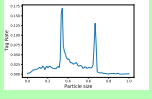
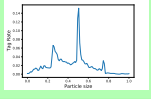
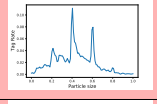
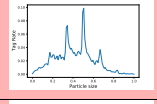
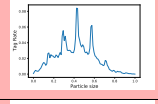
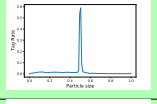
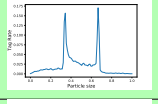
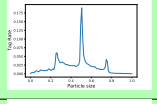
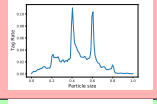
N	1 Memory	2 Memories	3 Memories	4 Memories	5 Memories	6 Memories
10 Particles						
50 Particles						
100 Particles						
500 Particles						
1000 Particles						
5000 Particles						
10000 Particles						
50000 Particles						
100000 Particles						

Table 1: *Attempts at writing various numbers of memories to systems with different numbers of particles. Background color of each cell is color coded to denote whether the attempt was successful or not, green for successful, red for unsuccessful. An attempt was only considered successful if all memories which were written are readable. All systems have the same particle density of n=0.01. For all systems* d_r *is equal to* $l_{box}/24000$*. All systems were trained to 30000 cycles with the smaller memories repeated. Area fractions were equally spaced within the interval (0, 1), i.e.* $af = 0.5$ *for one memory,* $af_1 = 0.6$ *and* $af_2 = 0.3$ *for two memories, and so on.* ϵ_{noise}*=0.001*

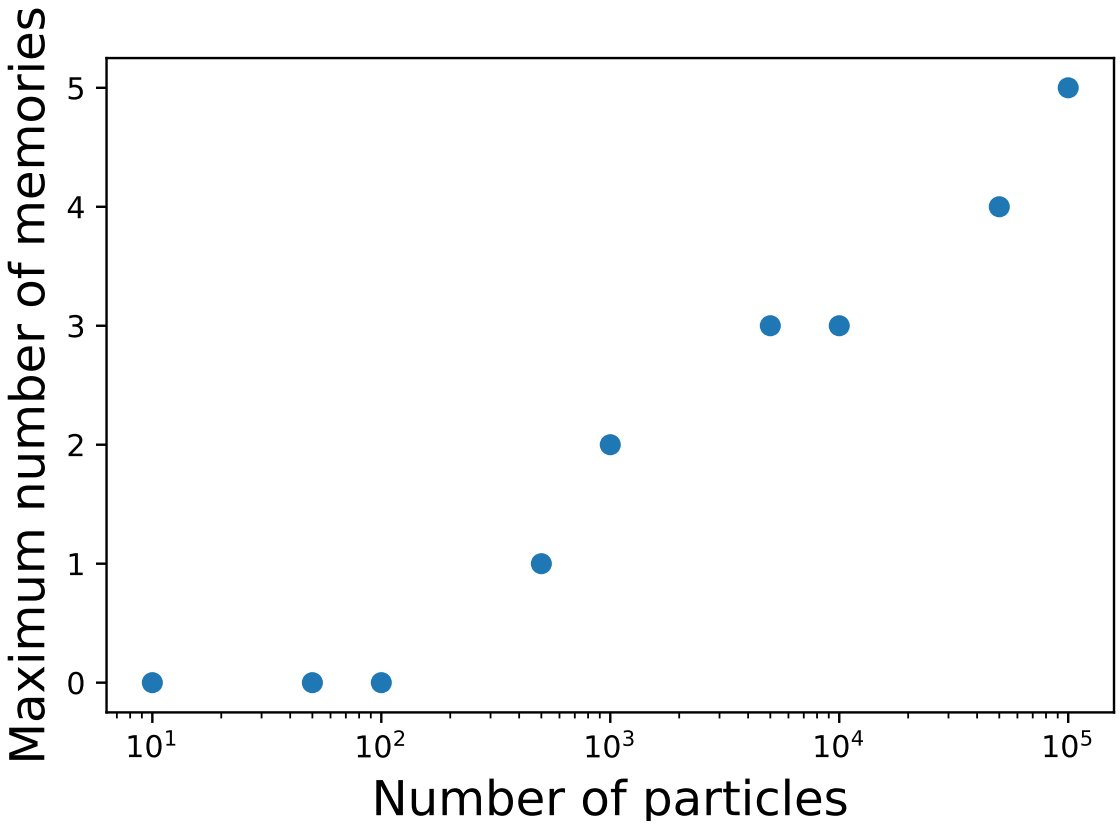


Figure 3: *Comparison of the memory readout for systems with different numbers of particles. For sufficiently small systems even at a single memory readout is impossible. As the number of particles increases the number of memories that can be read increases as well. The final plot displays the maximum number of readable memories vs the number of particles in the system.*

0.4 Particle Density

Fig. 4 illustrates the effect of particle number density n, which can be defined as the total number of particles divided by the total box area. At high density we observe that memories that are either too small or too large can no longer be written effectively. The line corresponding to the highest density does not have notable peaks at smaller or larger area fractions where more moderate densities do have peaks at these more extreme area fractions. At very low density, multiple memories can be written and read, like most other density values, but the peaks corresponding to individual memories become more distinct. As a result, having a lower number density of particles is preferable, simply for ease of reading the system for written memories.

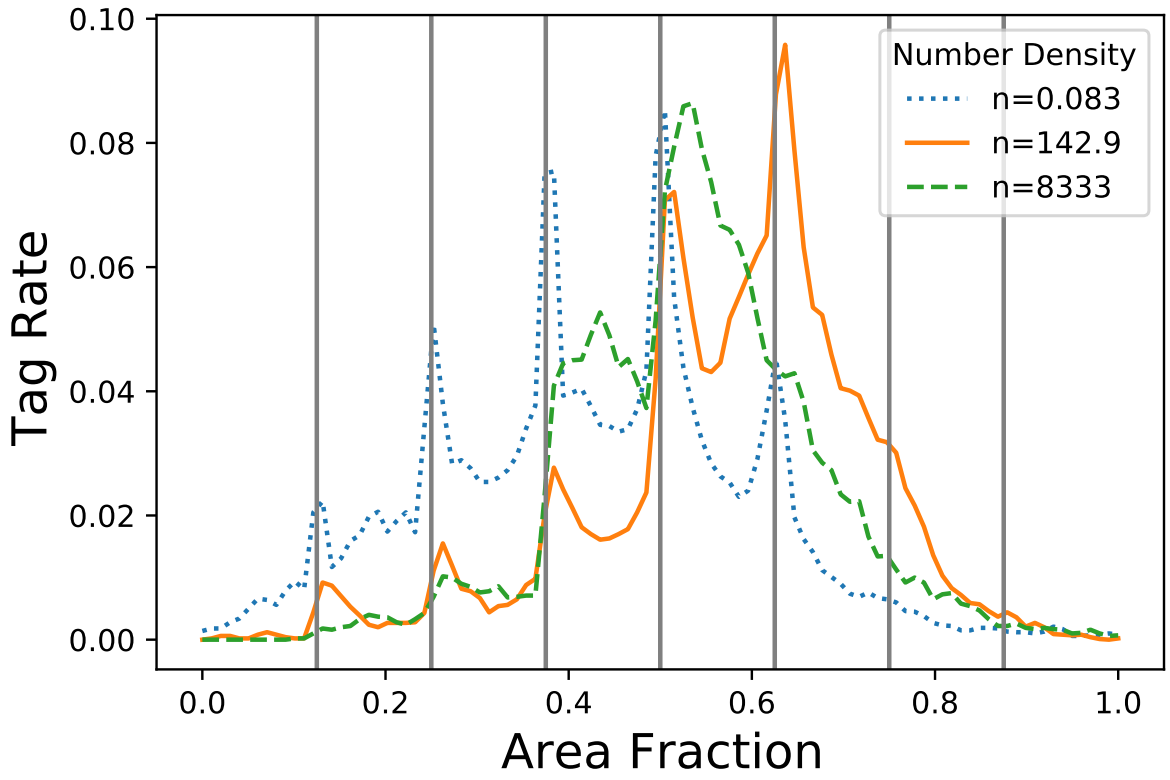


Figure 4: *Comparison of the memory readout for a wide range of particle number densities. For very low number densities the peaks for each memory are narrower and more distinct. For most number densities multiple memories are still clearly visible, but the peaks are broader and more blended. Only at very high number density do we loose the ability to write multiple memories. Kick is scaled with box size. Parameters used: N=10000, l_b scaled to meet desired density, d_r/l_b=1/24000, ϵ_{noise}=0.001.*

0.5 Kick

Fig. 5 illustrates the effect of the kick value. When paired with a high kick value, high number densities lead to memories not being written effectively above a given value. Low number densities alleviate this issue. This threshold only appears at large kick values. Due to the other effects of particle density, reducing the kick value appears to be the most appropriate way to resolve this.

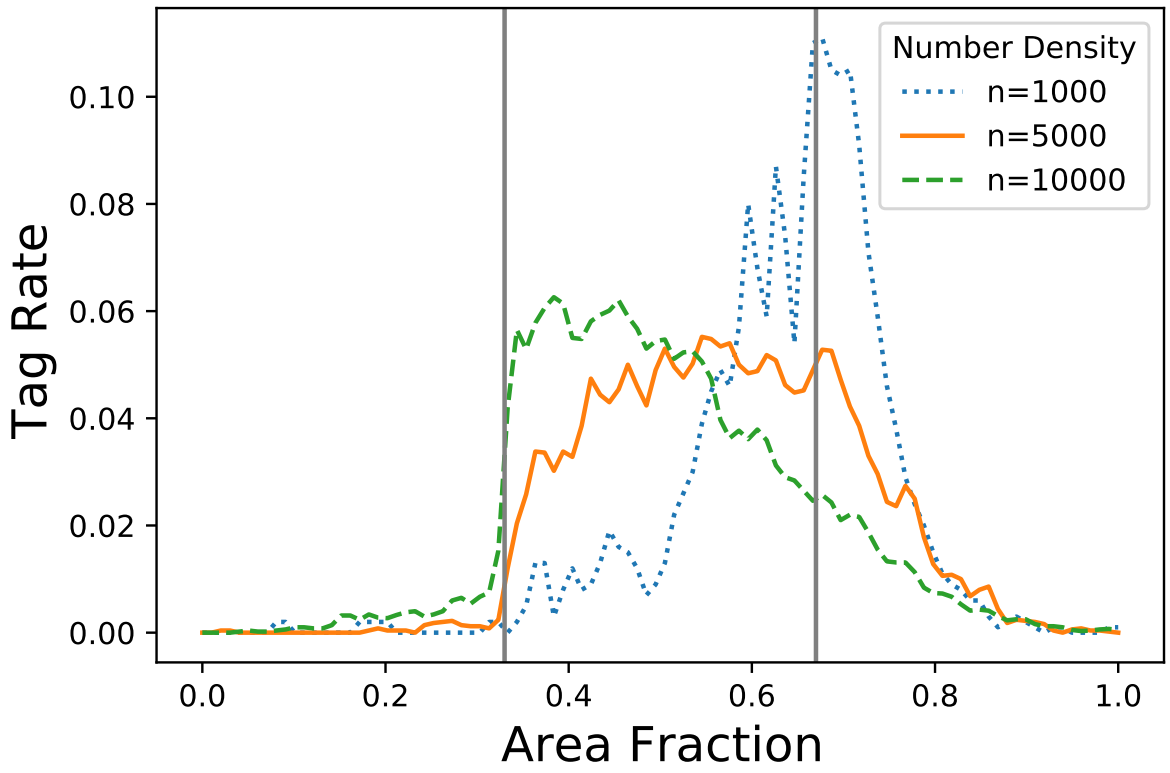


Figure 5: *Comparison of multiple particle number densities around the high kick threshold. High number densities lead to memories not being written effectively above a given value. Low number densities alleviate this issue. This threshold only appears at large kick values. Parameters used:* l_{box}*=12000,* d_r*=50,* ϵ_{noise}*=0.001*

Part IV
Discussion

This study investigates how the parameters of a system of swelling particles influence memory capacity.

The value of noise in the system is relevant only insofar as how the system behaves without noise and with a noise value that is too large. Without noise at long time the system will only retain the memory of the largest area fraction written. With too large a value of noise the system is not able to store memories and resembles an untrained system even at long time.

The number of particles in a system affects the clarity of the written memories. When writing multiple memories, a larger particle number leads to more distinguishability between the peaks corresponding to memories at different area fractions. Thus, the number of particles in a system can be understood as the resolution of the system, where more particles lead to a higher resolution. As such, if a system consists of too few particles, multiple memories can no longer be distinguished effectively. While in theory it would be desirable to use the largest number of particles possible, increasing N does have diminishing returns in improving the clarity, and also results in longer simulation run times.

The number density of the particles in the system appears to have very little impact on the writing and reading of multiple memories. Over density values ranging from $n = 16$ to $n = 833.3$ the plots of the particle tag rate appear qualitatively similar with the exception of the width of the peaks for each memory. Lower particle densities appear to lead to more distinct memories. This makes sense as the the number of tagged particles at different values of af with particles closer together should be more similar than if the particles were further apart.

The kick repels particles sufficiently far away from one another to no longer be tagged while reaching the trained area fraction. Noise drives particles toward the system's untrained state. When the kick value is too small, noise dominates and the tag rate plot begins to resemble that of the untrained state. Thus the kick and the noise are the primary influences on the system's training dynamics, in that they determine how fast the system progresses towards equilibrium, up until the kick becomes too large. The value at which d_r becomes too small is dependant on ϵ_{noise}, but once it reaches the threshold, the system resembles the high noise scenario depicted in Figure 1.

Rather than repelling particles gradually until they no longer interact, large values of d_r significantly increase the likelyhood that when two particles repel they are kicked nearer to other particles than before the kick. As a result, it becomes impossible for the system to write memories larger than a certain threshold, in our case observed around $af = 0.5$. This effect can be alleviated by decreasing the density of the particles, since then there are fewer particles near each other, and so there is less of a chance that the kick drives a given particle within swelling distance of a new particle. Based on the observations with regards to the effect of particle number density on the ability to read memories and the fact that the kick value appears to have little other effect on the system, it is likely preferable to change the kick value to be between the two thresholds observed rather than to change the density.

Part V
Conclusions

We have investigated the effects of a variety of parameters on a swelling particle system's ability to store multiple memories. The amount of noise in the system was found to have a notable effect. With a noise value that is too large, instead of writing multiple memories as intended, the system is constantly reverting to an untrained state. This is a result of particles having their positions randomly reassigned faster than they can be repelled, and so no memories are ever effectively written. Noise acts as a cap on how far toward the trained state the system can progress, with little to no noise allowing it to progress to where interactions no longer occur, and a large value of noise preventing the system from leaving a state statistically similar to no training having taken place.

The number of particles in the system mainly effects the resolution of the system, making the memories more distinct when read, and too few particles in a system makes it difficult to read memories at all. When the memories become more distinct, the memory capacity increases, since it is clearer when two adjacent peaks in the tag rate plot are in fact distinct memories rather than noise in the reading. The effect of particle number density is relatively small, but a lower number density does lead to more distinct memories when read. Unlike number of particles, where memory capacity increases because there is less noise in the tag rate plot, the memory capacity increase due to particle number density is a result of the peaks in the tag rate plot becoming narrower, and as a result, more can be fit into a smaller range. The effect of the kick on the storage of memories is notable in two distinct thresholds. A value of d_r that are too small, particles will not be able to repel sufficiently to store memories before noise interferes. Values of d_r that are instead too large result in a maximum area fraction.

References

[1] S. N. Coppersmith, T. C. Jones, L. P. Kadanoff, A. Levine, J. P. McCarten, S. R. Nagel, S. C. Venkataramani, and Xinlei Wu. Self-organized short-term memories. *Phys. Rev. Lett.*, 78:3983–3986, May 1997.

[2] Nathan C. Keim and Sidney R. Nagel. Generic transient memory formation in disordered systems with noise. *Phys. Rev. Lett.*, 107:010603, Jun 2011.

[3] Nathan C. Keim, Joseph D. Paulsen, and Sidney R. Nagel. Multiple transient memories in sheared suspensions: Robustness, structure, and routes to plasticity. *Phys. Rev. E*, 88:032306, Sep 2013.

[4] Joseph D. Paulsen, Nathan C. Keim, and Sidney R. Nagel. Multiple transient memories in experiments on sheared non-brownian suspensions. *Phys. Rev. Lett.*, 113:068301, Aug 2014.

[5] D. Pine, J. Gollub, J. Brady, and et al. Chaos and threshold for irreversibility in sheared suspensions. *Nature*, 438:997–1000, December 2005.