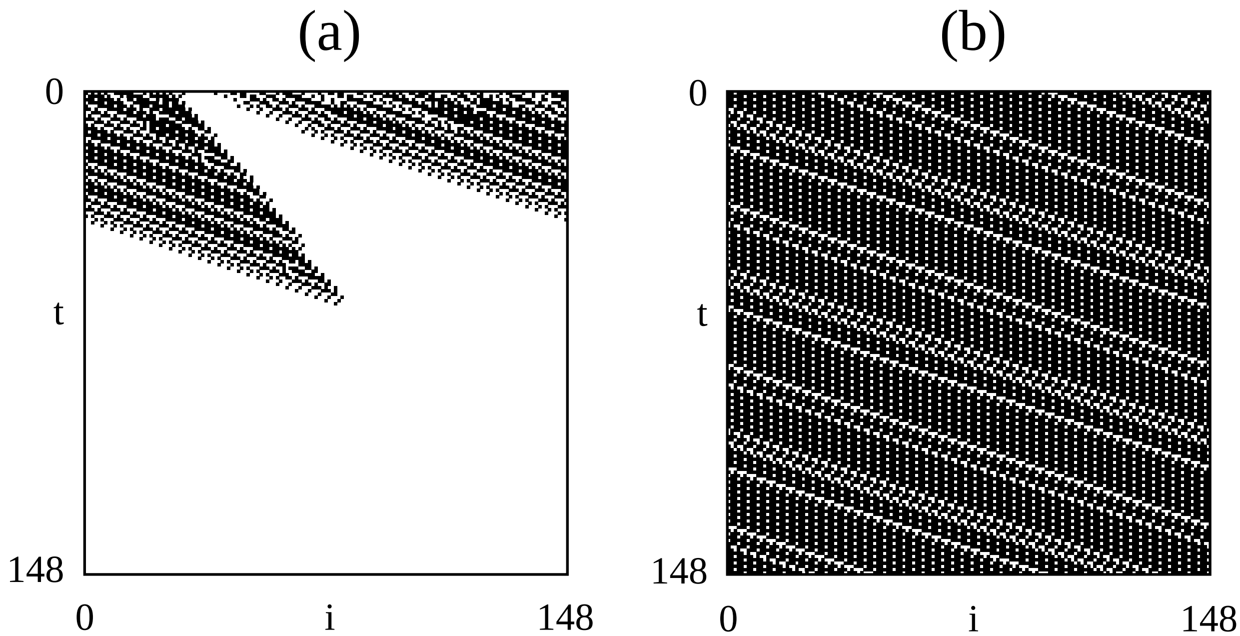

Figure 12: Space-time diagrams of one epoch-3 rule with $\lambda \approx 0.58$ that increases sufficiently large blocks of adjacent or nearly adjacent 0's. In (a) the initial configuration with $\rho \approx 0.42$ maps to a correct classification pattern of all 0's. In (b) the initial configuration with $\rho \approx 0.56$ is not correctly classified ($\rho(148) \approx 0.75$) but partial credit is given.

The general idea behind these two strategies is to rely on statistical fluctuations in the initial configurations. An initial configuration with $\rho > \rho_c$ is likely to contain a sufficiently large block of adjacent or nearly adjacent 1's. The rule then increases this region's size to yield the correct classification. Similarly, this holds for the CA in Figure 12 with respect to blocks of 0's in initial configurations with $\rho < \rho_c$. In short, these strategies are assuming that the presence of a sufficiently large block of 1's or 0's is a good predictor of $\rho(0)$.

Similar strategies were discovered in every run. They typically emerge by generation 20. A given strategy either increased blocks of 0's or blocks of 1's, but not both. These strategies result in a significant jump in fitness: typical fitnesses for the first instances of such strategies range from 0.75 to 0.85. This jump in fitness can be seen in the run of Figure 10 at approximately generation 10, and is marked as the beginning of epoch 3. This is the first epoch in which a substantial increase in fitness is associated with a symmetry breaking in the population. The symmetry breaking involves deciding whether to increase blocks of 1's or blocks of 0's. The GKL rule is perfectly symmetric with respect to the increase of blocks of 1's and 0's. The GA on the other hand tends to discover one or the other strategy, and the one that is discovered first tends to take over the population, moving the population λ's to one or the other side of 1/2. The causes of the symmetry breaking are explained below.

The first instances of epoch-3 strategies typically have a number of problems. As can be seen in Figures 11 and 12, the rules often rely on partial credit to achieve fairly high fitness on structurally incorrect classification. They typically do not get perfect scores on many initial configurations. The rules also often make mistakes in classification. Three common types of classification errors are illustrated in Figure 13. Figure 13(a) illustrates a

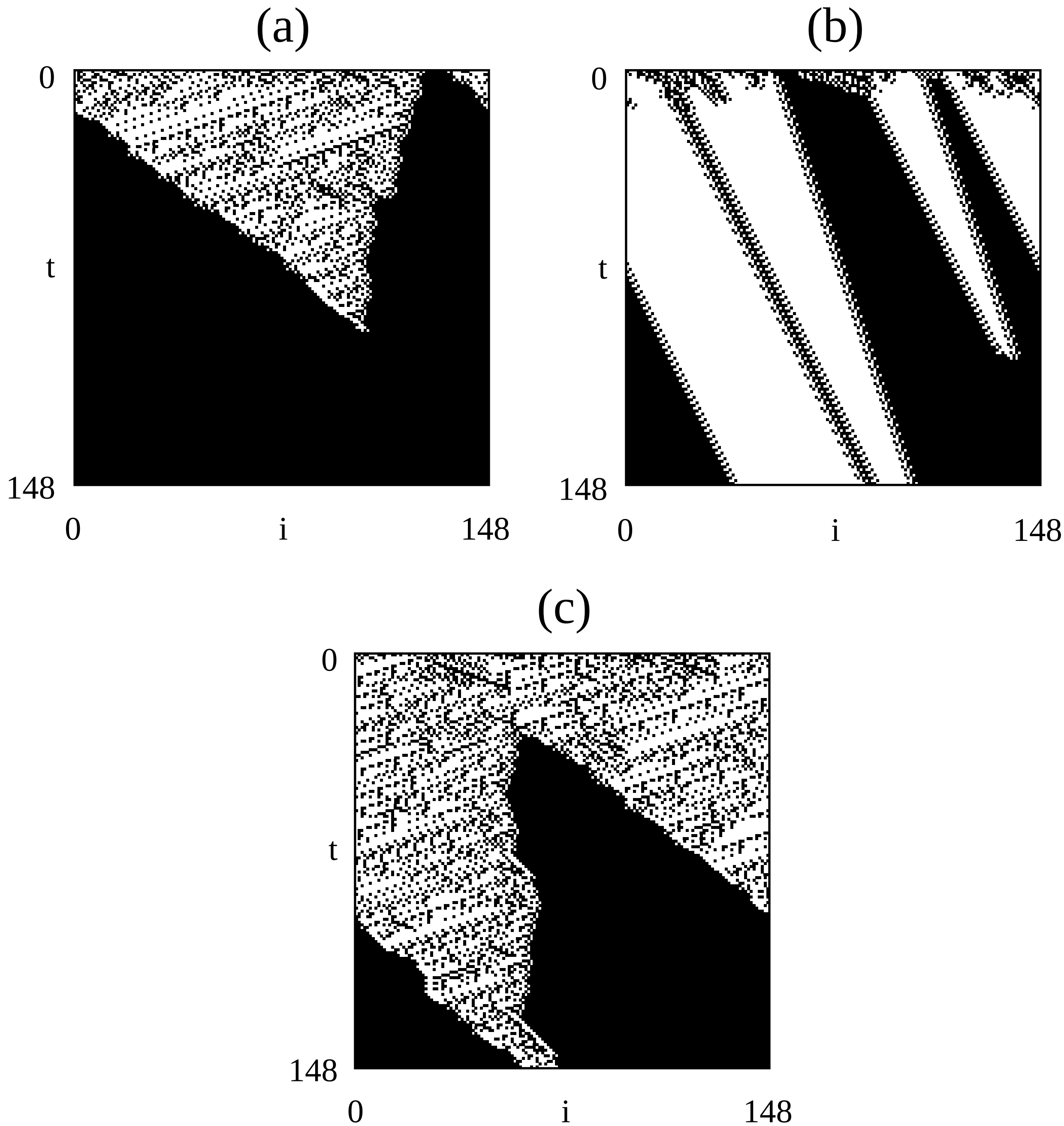

Figure 13: Space-time diagrams illustrating three types of classification errors committed by epoch-3 rules: (a) growing a block of 1s in a sea of $\rho < \rho_c$, (b) growing blocks of 1's for an initial configuration with $\rho > \rho_c$ too slowly (the correct fixed point of all 1's does not occur until iteration 480), and (c) generating a block of 1's from a sea of $\rho < \rho_c$ and growing it so that $\rho > \rho_c$ (the incorrect fixed point of all 1's occurs at iteration 180). The initial configuration densities are (a) $\rho(0) \approx 0.39$, (b) $\rho(0) \approx 0.59$, and (c) $\rho(0) \approx 0.45$.

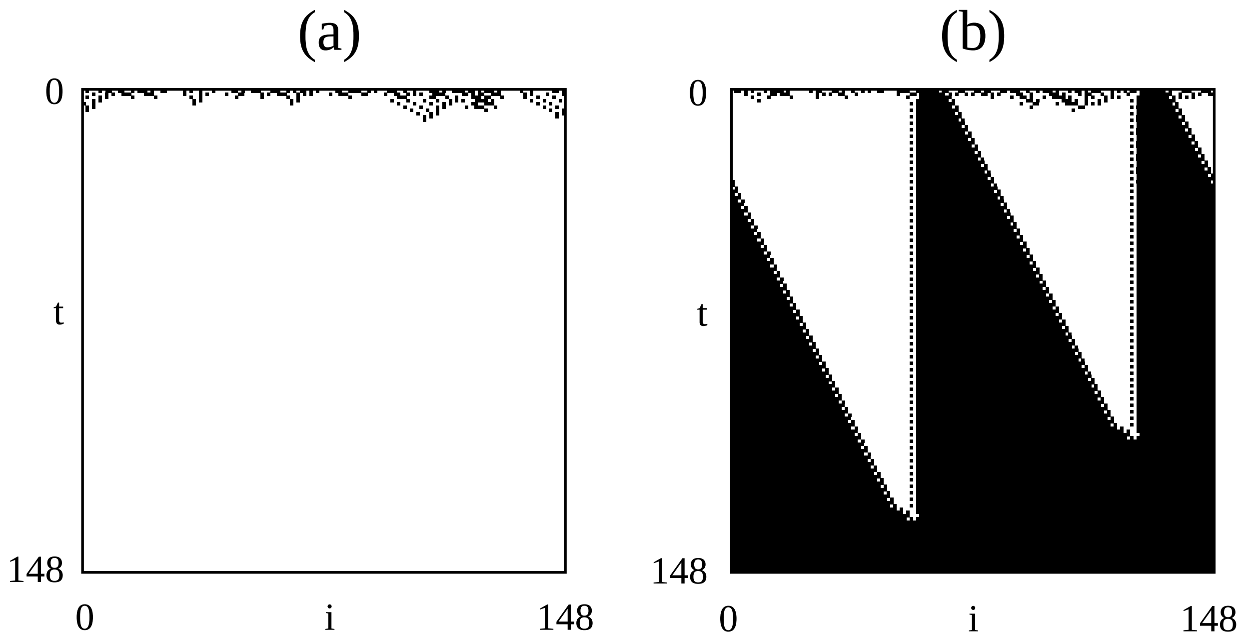

Figure 14: Space-time diagrams of one epoch-4 rule with $\lambda \approx 0.38$ that increases sufficiently large blocks of adjacent or nearly adjacent 1's. In (a) $\rho(0) \approx 0.44$; in (b) $\rho(0) \approx 0.52$. Both initial configurations are correctly classified.

rule increasing a too-small block of 1's and thus misclassifying an initial configuration with $\rho < \rho_c$. Figure 13(b) illustrates a rule that does not increase blocks of 1's fast enough on an initial configuration with $\rho > \rho_c$, leaving many incorrect bits in the final lattice. Figure 13(c) illustrates the *creation* of a block of 1's that did not appear in an initial configuration with $\rho < \rho_c$, ultimately leading to a misclassification. The rules that produced these diagrams come from epoch 3 in various GA runs.

The increase in fitness seen in Figure 10 between generation 10 and 20 or so is due to further refinements of the basic strategies that correct these problems to some extent.

Epoch 4: Reaching and staying at a maximal fitness

In most runs, the best fitness is typically at its maximum value of 0.90 to 0.95 by generation 40 or so. In Figure 10 this occurs at approximately generation 20, and is marked as the beginning of epoch 4. The best fitness does not increase significantly after this; the GA simply finds a number of variations of the best strategies that all have roughly the same fitness. When we extended 16 of the 30 runs to 300 generations, we did not see any significant increase in the best fitness.

The actions of the best rules from generation 100 of two separate runs are shown in Figures 14 and 15. The leftmost space-time diagrams in each figure are for initial configurations with $\rho < \rho_c$, and the rightmost diagrams are for initial configurations with $\rho > \rho_c$. The rule illustrated in Figure 14 has $\lambda \approx 0.38$; its strategy is to map initial configurations to 0's unless there is a sufficiently large block of adjacent or nearly adjacent 1's, which if present is increased. The rule shown in Figure 15 has $\lambda = 0.59$ and has the opposite strategy. Each of these rules has fitness ≈ 0.93. They are better tuned versions of the rules in Figures 11

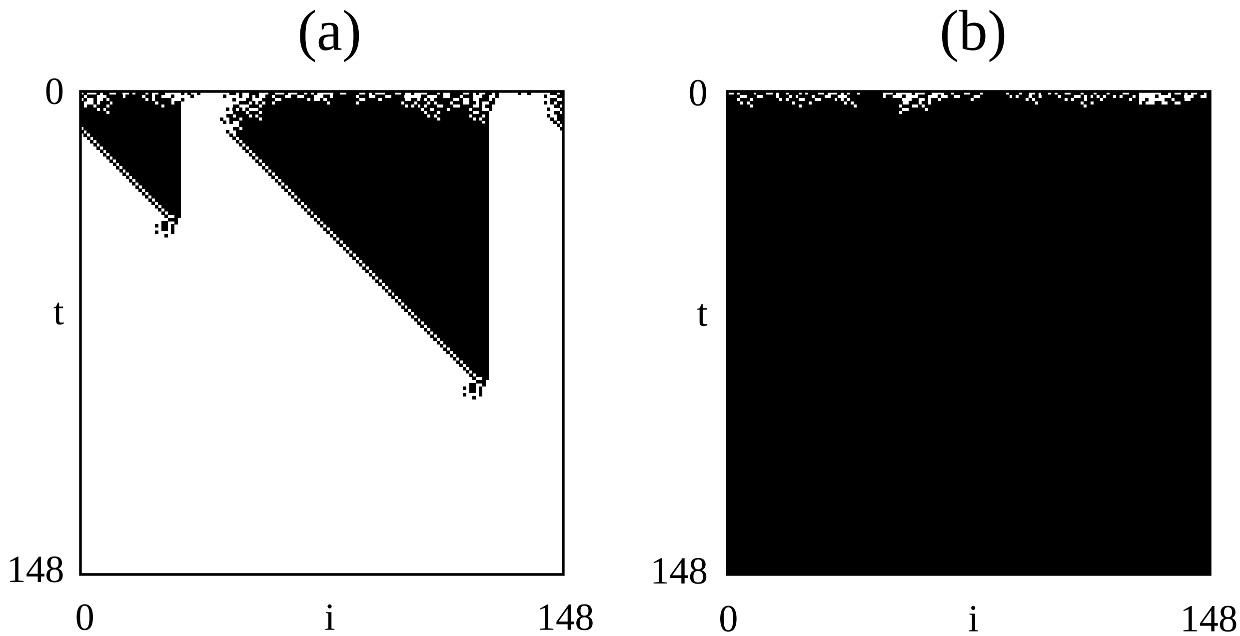

Figure 15: Space-time diagrams of one epoch-4 rule with $\lambda \approx 0.59$ that increases sufficiently large blocks of adjacent or nearly adjacent 0's. In (a) $\rho(0) \approx 0.40$; in (b) $\rho(0) \approx 0.56$. Both initial configurations are correctly classified.

and 12.

Symmetry breaking in epoch 3

Notice that the λ values of the rules that have been described are in the bins centered around 0.43 and 0.57 rather than 0.5. In fact, it seems to be much easier for the GA to discover versions of the successful strategies close to $\lambda = 0.43$ and $\lambda = 0.57$ than to discover them close to $\lambda = 1/2$, though some instances of the latter rules were found. Why is this? One reason is that rules with high or low λ work well by *specializing*. The rules with low λ map most neighborhoods to 0's and then increase sufficiently large blocks of 1's when they appear. Rules with high λ specialize in the opposite direction. A rule at $\lambda = 1/2$ cannot easily specialize in this way. Another reason is that a successful rule that grows sufficiently large blocks of (say) 1's must avoid *creating* a sufficiently large block of 1's from an initial configuration with less than half 1's. Doing so will lead it to increase the block of 1's and produce an incorrect answer, as was seen in Figure 13(c). An easy way for a rule to avoid creating a sufficiently large block of 1's is to have a low λ. This ensures that low-density initial configurations will quickly map to all 0's, as was seen in Figure 14(a). Likewise, if a rule increases sufficiently large blocks of 0's, it is safer for the rule to have a high λ value so it will avoid creating sufficiently large blocks of 0's where none existed. A rule close to $\lambda = 1/2$ will not have this safety margin, and may be more likely to inadvertently create a block of 0's or 1's that will lead it to a wrong answer. A final element that contributes to the difficulty of finding good rules with $\lambda = 1/2$ is the combinatorially large number of rules there. In effect, the search space is much larger, which makes the global search more difficult. Locally, about a given adequate rule at $\lambda = 1/2$, there are many more rules close in Hamming distance and thus reachable via mutation that are not markedly better.

Once the more successful versions of the epoch-3 strategies are discovered in epoch 4, their variants spread in the population, and the most successful rules have λ on the low or high side of $\lambda = 1/2$. This explains the shift from the clustering around $\lambda = 1/2$ as seen in generations 10–30 in Figure 9 to a two-peaked distribution that becomes clear around generation 65. The rules in each run cluster around one or the other peak, specializing in one or the other way. We believe this type of symmetry breaking may be a key mechanism that determines much of the population dynamics and the GA's success—or lack thereof—in optimization.

How does this analysis of the symmetry breaking jibe with the argument given earlier that the best rules for the $\rho_c = 1/2$ task must be close to $\lambda = 1/2$? None of the rules found by the GA had a fitness as high as 0.98—the fitness of the GKL rule, whose λ is exactly 1/2. That is, the evolved rules make significantly more classification errors than the GKL rule, and, as will be seen below, the measured fitness of the best evolved rules is much worse on larger lattice sizes, whereas the GKL rule's fitness remains roughly the same across lattice sizes. To obtain the fitness of the GKL rule a number of careful balances in the rule table must be achieved. This is evidently very hard for the GA to do, especially in light of the symmetries in the task and their suboptimal breaking by the GA.

7.5 Performance of the Evolved Rules

Recall that the proportional fitness of a rule is the fraction of correct cell states at the final time step, averaged over 300 initial configurations. This fitness gives a rule partial credit for getting some final cell states correct. However, the actual task is to relax to either all 1's or all 0's, depending on the initial configuration. In order to measure how well the evolved rules actually perform the task, we define the *performance* of a rule to be the fraction of times the rule correctly classifies initial configurations, averaged over a large number of initial configurations. Here, credit is given only if the initial configuration relaxes to exactly the correct fixed point after some number of time steps. We measured the performance of each of the elite rules in the final generations of the 30 runs by testing it on 300 randomly generated initial configurations that were uniformly distributed in the interval $0 \leq \rho \leq 1$, letting the rule iterate on each initial condition for 1000 time steps. Figure 16 displays the mean performance (diamonds) and best performance (squares) in each λ bin. This figure shows that while the mean performances in each bin are much lower than the mean fitnesses for the elite rules shown in Figure 6, the best performance in each bin is roughly the same as the best fitness in that bin. (In some cases the best performance in a bin is slightly higher than the best fitness shown in Figure 6. This is because different sets of 300 initial conditions were used to calculate fitness and performance. This difference can produce small variations in the fitness or performance values.) The best performance we measured was ≈ 0.95. Under this measure the performance of the GKL rule is ≈ 0.98. Thus the GA never discovered a rule that performed as well as the GKL rule, even up to 300 generations. In addition, when we measure the performance of the fittest evolved rules on larger lattice sizes, their performances decrease significantly, while that of the GKL rule remains roughly the same.