# Controlling State-Space Explosion in Chunk Learning

**Vladislav D. Veksler**, **Kevin A. Gluck**, **Christopher W. Myers**, **Jack Harris**, **Thomas Mielke**
Air Force Research Laboratory, Wright-Patterson AFB, USA

**Abstract**

Varying combinations of perceptual cues may be relevant for learning and action-selection. However, storing all possible cue combinations in memory is computationally implausible in sufficiently complex environments due to a state-space explosion. Some psychological models suggest that cue combinations, i.e. chunks, should be generated at a conservative rate (EPAM/CHREST; e.g. Feigenbaum & Simon, 1984). Other models suggest that chunk retrieval is based on statistical regularities in the environment (i.e. recency and frequency; Anderson & Schooler, 1991). We present a computational model of chunk generation based on these two principles, and demonstrate how combining these principles alleviates state-space explosion, producing great savings in memory while maintaining a high level of performance.

**Keywords:** chunking, unitization, configural-cue, state-space explosion, combinatoric explosion, learning, rational analysis

## Introduction

A brown apple is neither just brown, nor just an apple, nor just a brown apple. It is all of these things simultaneously, and any of these representations may be important for both learning and action-selection. Concurrent representation of all cue combinations allows for learning of both generic rules (e.g. apples taste good) and exceptions to those rules (e.g. brown apples are spoiled).

In the psychological literature a set of perceptual cues is known as a chunk (e.g. Feigenbaum & Simon, 1984) or a configural-cue (Wagner & Rescorla, 1972). Having a single memory chunk for representing a given combination of cues allows for faster recognition of current state and faster action selection (e.g. Goldstone, 2000). Chunk representation also aids in the ability to store greater amounts of information in working memory (Ericsson, Delaney, Weaver, & Mahadevan, 2004). A commonly cited example of this phenomenon involves improved recognition of displayed chess positions by chess experts compared to novices (Chase & Simon, 1973; Gobet, 1998). The theory is that experts will have been exposed to more chess situations, and will have learned many chunks for representing complex states on the chess board. Thus a chess expert will need fewer chunks to represent a chess position than a novice, who may have to memorize the position as a series of figures and their locations. This theory is further supported by the fact that experts are no better than novices in recall of chess boards where pieces are placed randomly (rather than positions encountered during actual game-play).

Each set of perceptual cues may potentially be recognized as multiple concurrently active chunks. Cues *large*, *square*, and *white* may activate seven different chunks: {*large*}, {*square*}, {*white*}, {*large*, *square*}, {*large*, *white*}, {*square*, *white*}, and {*large*, *square*, *white*}. Recognition of a set of perceptual cues as multiple concurrently active chunks can account for a wide range of behavioral phenomena, including classical conditioning (Pearce, 1994), base-rate neglect (Gluck & Bower, 1988a, 1988b), and categorization (e.g. Gluck & Bower, 1988b; Veksler, Gray, & Schoelles, 2007).

The problem with a full chunk representation, where each set of cues activates all possible combinations of those cues in memory, is that in complex environments too many chunks may be required. A mere ten binary perceptual inputs (e.g. black vs white, large vs small) may require over 59 thousand chunks to be present in memory[1]. If each perceptual input allowed for five possible values (e.g. black, dark-grey, grey, light-grey, white), ten such input dimensions could result in almost ten million chunks. Twenty such perceptual dimensions would result in 95 trillion chunks. One hundred inputs with ten values per input would result in more chunks than there are atoms in the observable universe. Given these numbers, it is safe to say that the brain does not create a memory element for each combination of input signals, and a computational system based on a full chunk representation is not feasible in complex environments.

The exponential growth of memory based on combinations of perceptual cues is called the state-space explosion problem. This problem becomes even more severe when a chunk-based memory system is integrated with State-Action and/or State-State connections (e.g. Gluck & Bower, 1988b; Pearce, 1994; Veksler et al., 2007), where the total amount of required memory may be an exponential function of the number of chunks, creating double-exponential memory growth.

There is much psychological evidence that humans do not hold memory representations for all potential combinations of situational features. Rather, cues get merged over time to make more and more complex object representations. There is evidence of this phenomenon in the context of perceptual unitization within hour-long psychological experiments (e.g. Goldstone, 2000), as well as in life-long acquisition of subject-matter expertise (e.g. Feigenbaum & Simon, 1984).

In this paper we present a model of gradual chunk learning based on widely supported principles of human memory, and examine how this helps to alleviate the state-space

[1]Given $n$ cues (e.g. *large*, *square*, *white*), we can create a chunk for every combination of cue presence and absence ({*large*}, {*square*}, {*white*}, {*large*, *square*}, {*large*, *white*}, {*square*, *white*}, and {*large*, *square*, *white*}). If we represent cue presence as a 1 and cue absence as a 0, we can represent each chunk as a binary number, and the total number of possible chunks is the total number of possible binary numbers (minus the blank chunk), which is $2^n - 1$. When each cue dimension can have two potential values, the total number of possible chunks is $3^n - 1$. With $k - 1$ possible values on $n$ feature dimensions, we can have at most $k^n - 1$ possible chunks to represent all potential cue combinations.

explosion. Specifically, we propose that state-space explosion is mitigated if new chunks are created via the union of existing memory chunks (1) only when none of the existing memory chunks fully represent the current state (similar to *familiarization* in EPAM/CHREST; e.g. Feigenbaum & Simon, 1984; Gobet et al., 2001), and (2) only when the activation of existing chunks goes above threshold, where activation is based on statistical regularities in the environment (i.e. recency and frequency; Anderson & Lebiere, 1998; Anderson & Schooler, 1991). We present two simulations, demonstrating how combining these two principles alleviates state-space explosion, producing significant memory savings while maintaining a high level of performance. The simulations show that the first of these principles slows down chunk generation, although growth remains a linear function of time; whereas the second principle aids in more selective "rational" chunk generation, where growth is actually slower in highly variable environments with more potential stimuli.

## Gradual Chunk Learning via the Conservative-Rational principles of memory

According to Simon's early computational models of learning (EPAM; e.g. Feigenbaum & Simon, 1984), agent state may be represented as a chunk – a mathematical set of perceptual cues (e.g. $\{large, white, square\}$), and larger, more specific chunks are generated by adding more cues to existing chunks (i.e. *familiarization* or *chunking*; $\{large, white, square\} \cup \{rotated\} = \{large, white, square, rotated\}$). Familiarization in EPAM happens gradually, and only when the model does not yet contain a chunk that includes the entire set of perceptual cues in the observed state. That is, early on states are represented in memory as individual cues, then as small chunks, and then, gradually, as chunks of greater and greater specificity.[2]

Anderson's work on memory, in turn, suggests that chunks can be retrieved when their activation is above threshold. Chunk activation is based on the statistical properties of the environment (rational analysis; Anderson & Lebiere, 1998; Anderson & Schooler, 1991). If the set of cues that make up a chunk co-occur in recent history, and/or co-occur frequently, this chunk is deemed relevant, and thus more likely to be retrieved.

Combining the *conservative chunk growth* and the *rational analysis* principles together, we propose a conservative-rational chunk learning model. In general, the model may be described as such: (1) new chunks are generated gradually via the union of retrieved existing memory chunks only when none of the existing chunks fully represent the current state, (2) chunks are retrievable only when their activation level goes above threshold, (3) the more active chunks are more likely to be retrieved, and (4) chunk activation level is based on the recency and frequency of chunk occurrence in the environment (or, rather, the occurrence of the cues that make up that particular chunk).

More formally, when the conservative-rational model is initiated it contains one base-level chunk for every perceptual cue that may be observed in the environment, such that if there are $x$ possible cues, there are also $x$ unique chunks in memory, $M$, each containing its respective perceptual cue, $M = \{\{cue_1\}...\{cue_x\}\}$. When the model is exposed to some perceptual input containing a set of cues, $F$, activation is added to each existing memory chunk in set:

$$S = \{s | s \in M, s \subseteq F\}$$

Chunk activation decays over time. At any point the activation of chunk $i$, $A_i$, can be calculated as follows (vis-a-vis base-level activation formula from ACT-R; Anderson & Lebiere, 1998):

$$A_i = \sum_{j=1}^{n} t_j^{-d} + \sigma$$

where $n$ is the number of presentations of chunk $i$, $t_j$ is the time elapsed since the $j$th presentation of $i$ (in steps; $t_j \geq 1$)[3], $d$ is the decay rate (for all simulations below decay rate is set to 1.0, which is equivalent to hyperbolic decay; different decay rates would not qualitatively alter simulation results), and $\sigma$ is noise (a small random number added to chunk activation to add stochasticity to the model).

At each step the model attempts to retrieve two chunks, $a$ and $b$, so as to add a new chunk to memory, $M = M \cup \{a \cup b\}$, where $a$ and $b$ are the most active candidates from the set of chunks where no chunk is a subset of any other active chunk, $\{i \in S | \forall_{j \in S} i \not\subset j\}$ (e.g. if the set of activated chunks is {{white}, {large}, {rotated}, {oval}, {white, large}, {white, large, rotated}}, then {white, large, rotated} and {oval} will be the only candidates for creating a new chunk). Chunks $a$ and $b$ are retrievable only if their activations, $A_a$ and $A_b$, are both above the retrieval threshold parameter, $\rho$.

Note that for all chunks $s \in S$ the most recent elapsed time $t_j$ is 1, and $A_s >= 1$. Thus, when $\rho <= 1$ this model is just conservative and not rational, since every chunk in $S$ is above threshold, regardless of recency and frequency of prior activations.

The following simulations demonstrate how the conservative-rational principles of memory can help to alleviate the state-space explosion.

## Simulation 1: Alleviating state-space explosion

We ran the proposed chunk-learning model for 100,000 steps in environments with five and ten input dimensions, varying

[2]EPAM/CHREST models actually replace smaller chunks with larger ones during the familiarization process. An additional learning process, discrimination, is used to recreate smaller chunks if these are deemed needed later. For current purposes we borrow only the EPAM chunk generation principles, and not the processes of chunk deletion and recreation, as these are not compatible with the Reinforcement Learning model employed in Simulation 2.

[3]This model implementation is event-based, counting each perception-action cycle as a discrete step rather than in continuous time. Assuming constant cycle times, these are computationally equivalent. Elapsed time begins at 1, because at $t_j = 0$, $t_j^{-d} = \infty$

the retrieval threshold parameter, $\rho$ between 1.0 (no recency-frequency effect) to 2.5. The results, displayed as averages from ten model runs and compared to a model that created all possible chunks, are displayed in Figure 1. For both five- and ten- dimensional inputs we examined chunk growth for static, constrained, and chaotic environments.

The static environment simulation consisted of repeatedly exposing the model a set of five cues in a five-dimensional environment (e.g. step 1: $\{a, b, c, d, e\}$, step 2: $\{a, b, c, d, e\}$, ... step 100,000: $\{a, b, c, d, e\}$), and ten cues in a ten-dimensional one. In such an environment there are 31 possible chunks to be created with 5 input dimensions, and 1023 possible chunks to be created with 10 input dimensions. However, the proposed model (at all $\rho$ values) creates only 9 and 19 chunks in the 5- and 10- dimensional environments, respectively.

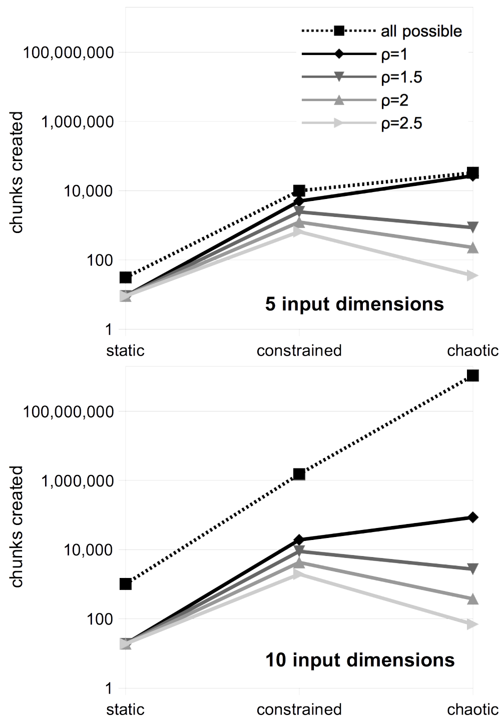

*Figure 1.* Chunks created after 100,000 steps in static, constrained, and chaotic 5-dimensional and 7-dimensional environments. Standard error is negligible.

This happens because chunks in the conservative-rational model are created only when necessary. Thus, in the 5-dimensional static environment where the model is constantly presented the state $\{a, b, c, d, e\}$, the model will be initiated with five chunks – $\{a\}$,$\{b\}$,$\{c\}$,$\{d\}$,$\{e\}$ – and will create only four more chunks (for example, it might create chunks in the following sequence $\{b, c\}$, $\{b, c, e\}$, $\{a, b, c, e\}$, and $\{a, b, c, d, e\}$). No other chunks will be generated because the chunk $\{a, b, c, d, e\}$ is sufficient to represent the state $\{a, b, c, d, e\}$.

In the chaotic environment the value of each dimensions could take on one of seven unique values, or could be absent from the input state (with a minimum of two cues present at each step). At each step the number of cues in the state was chosen at random, with a minimum value of 2 and a maximum value of 5 or 10 in the 5-dimensional and 10-dimensional environments, respectively. The cue value for each respective cue dimension was a number between 1 and 7, chosen at random, with replacement. That means any combinations of length greater than 2 had the potential to be presented in this environment (e.g. in the 5-dimensional environment the states could be $\{a : 1, b : 1, c : 2\}$, $\{a : 7, b : 1\}$, $\{b : 4, c : 5, d : 7, e : 7\}$).

A maximum of 32,767 chunks may be created in the 5x7 chaotic environment, and 1,073,741,823 chunks in the 10x7 one. The $\rho = 1$ model (no recency-frequency effects) generates on average 26,895 and 85,841 chunks in the 5x7 and 10x7 environments, respectively. Note that in the 10x7 environment the $\rho = 1$ model generates chunks at a pace of almost one per step. When the retrieval threshold is above 1.0, chunks are generated at a much more conservative pace. At $\rho = 1.5$ the conservative-rational model creates on average 856 and 2,736 chunks in the 5x7 and 10x7 environments, respectively. At $\rho = 2.0$ the model creates on average 228 and 380 chunks in the 5x7 and 10x7 environments, respectively. At $\rho = 2.5$ the model creates *no chunks* beyond the initial 35 and 70 chunks in the 5x7 and 10x7 environments, respectively.

Finally, the constrained environment was a simulated random walk in a room with various furniture by a robot with 5 proximity sensors (at 0°, 10°, −10°, 45°, −45°) or 10 proximity sensors (at 2°, −2°, 10°, −10°, 30°, −30°, 45°, −45°).[4] Each proximity sensor reading was converted to an integer between 1 and 7. In this way the room walk had the potential to have similar complexity to the 5x7 and 10x7 chaotic environments, except that it was constrained by actual simulated robot movements. The room layout is displayed in Figure 2.

In the simulated room walk, the all-chunks model generates a total of 9,928 chunks and 1,528,477 chunks for 5-sensor and the 10-sensor robots, respectively. The $\rho = 1$ model (no recency-frequency effects) generates on average 4,988 and 19,183 chunks in the 5-sensor and 10-sensor environments, respectively. At $\rho = 1.5$ the conservative-rational model creates on average 2,430 and 8,947 chunks in the 5-sensor and 10-sensor environments, respectively. At $\rho = 2$ the model creates on average 1,238 and 4,293 chunks in the 5-sensor and 10-sensor environments, respectively. At $\rho = 2.5$ the model creates on average 652 and 1,961 chunks in the 5-sensor and 10-sensor environments, respectively.

Note that as the environment becomes more complex, from static to constrained to chaotic, the number of chunks created in the all-chunks and $\rho = 1$ models increases. However, the $\rho > 1$ (recency-frequency) models create fewer chunks in the chaotic than in the constrained environments. When the world makes some sense, the principles of rational analysis

[4] All models in the 5-sensor environment were exposed to the same random walk, and all models in the 10-sensor environment were exposed to the same random walk.

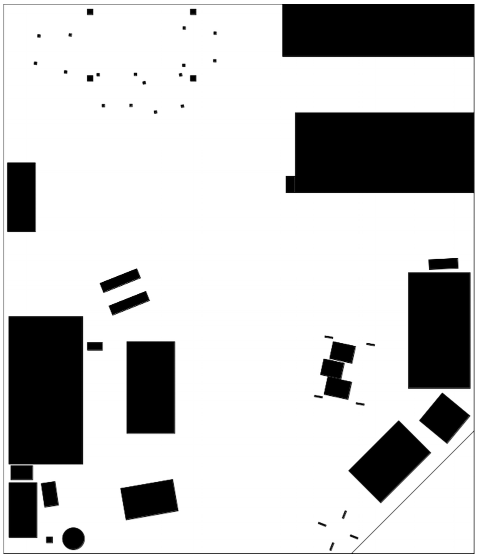

*Figure 2*. Simulated room environment used in Simulation 1 (drawn to scale from an actual room). All filled areas represent obstacles at the robot sensor height (e.g. the twenty dots at the top-left of the figure are chair and table legs).

help to derive which world features co-occur frequently and continuously in the environment, whereas when the world is pure noise, these principles suggest that allocating memory and processing to various cue combinations is not warranted. In other words, rational analysis helps to derive signal from noise, and allocate memory only to the most relevant cue combinations.

This type of "rational" chunk generation is much different than merely creating chunks at a conservative rate. Even if chunks are only created once every few seconds, or only a small proportion of the time, chunk growth would be a linear function of time, creating as many chunks from incidental cue co-occurrences in the chaotic environment as from the co-occurring features in the more ecologically-constrained environment. Alternatively, the recency/frequency principles provide an informed approach to chunk learning, based on the likelihood of chunk occurrence. Thus, in noisier environments, where the potential for state-space explosion is greatest, the proposed model creates fewer chunks.

## Simulation 2: But does it work?

The above simulation demonstrates how the conservative-rational principles of memory help to mitigate the state-space explosion in chunk learning, producing as much as 99.99% memory savings. An important question, however, is whether such a low number of chunks can produce a level of performance similar to a model that generates all possible chunks. To examine this, we integrated chunk-based memory with a Reinforcement Learning action-selection mechanism (Sutton & Barto, 1998), and examined model performances in a binary decision environment (for more detail as to how chunk-based memory is integrated with Reinforcement Learning, please see Veksler et al., 2007).

The environment generator we used for this simulation was developed to examine various binary decision environments (e.g. to send a patient to CCU or not; to classify network activity as a cyber-attack or not). The generator creates decision cases based on different binary cue distributions and outcome rules. We used the generator to create an environment approximating a generic threat detection task. On each trial in this environment each model had to classify a set of 10 or 12 binary perceptual cues as either a threat or not a threat. Each model had to operate and adapt through four consecutive threat scenarios, each scenario lasting 500 trials. In the first scenario the threat could be identified via a combination of three randomly chosen perceptual cue values (e.g. when patient has red spots, and coughing, and reporting chest pain, classify as a threat). In the second, third, and fourth scenarios the threat could be identified via combinations of four, three, and four randomly chosen perceptual cue values, respectively. The probability of a threat on any given trial was 50%.[5]

The reward values for correctly classifying a threat and missing a threat were relatively high (+10 and -10, respectively), the reward value for correctly rejecting a given scenario as a valid threat was relatively low (+1), the value of sounding a false alarm was somewhere in between those two (-5).[5] The models were evaluated based on the average reward per trial that they were able to achieve during the simulation.

Figure 3 displays performance and number of chunks created in this simulation for five models: (1) a model with all possible chunks, (2) the standard instance-based Reinforcement Learning model, where each distinct combination of input values was treated as a unique state, (3) a feature-based model, where no new chunks were created beyond the initial 20 chunks in the 10-dimensional and 24 chunks in the 12-dimensional decision environments (one chunk for each potential cue value), (4) a chunk learning model that employs no recency-frequency principles ($\rho = 1$), and (5) the full conservative-rational model ($\rho = 1.5$).

In the 10-dimensional environment the performance rank of the five models was as follows: the all-chunks model performed best (average reward of 5.2 per trial over 2000 trials), followed by the $\rho = 1$ model (-10.6%), then $\rho = 1.5$ (-13.2%), feature-based (-28.6%), and instance-based models (-52.7%). In the 12-dimensional environment the performance rank of the five models was as follows: the all-chunks model performed best (average reward of 5.0 per trials over 2000 trials), followed by the $\rho = 1.5$ model (-13.3%), then $\rho = 1$ (-13.8%), feature-based (-31.4%), and instance-based models (-91.1%).

All models, except the all-chunks model, performed better in the second half of the simulation, as time was needed for training and chunk generation. In the second half of the simulation performance shortfalls for the $\rho = 1$, $\rho = 1.5$, feature-based, and instance-based models were -8.4%, -9.6%,

[5]Simulation parameters (e.g. threat probabilities, reward values) are reported here for purposes of replicability. Main result trends do not vary greatly with different simulation parameters.

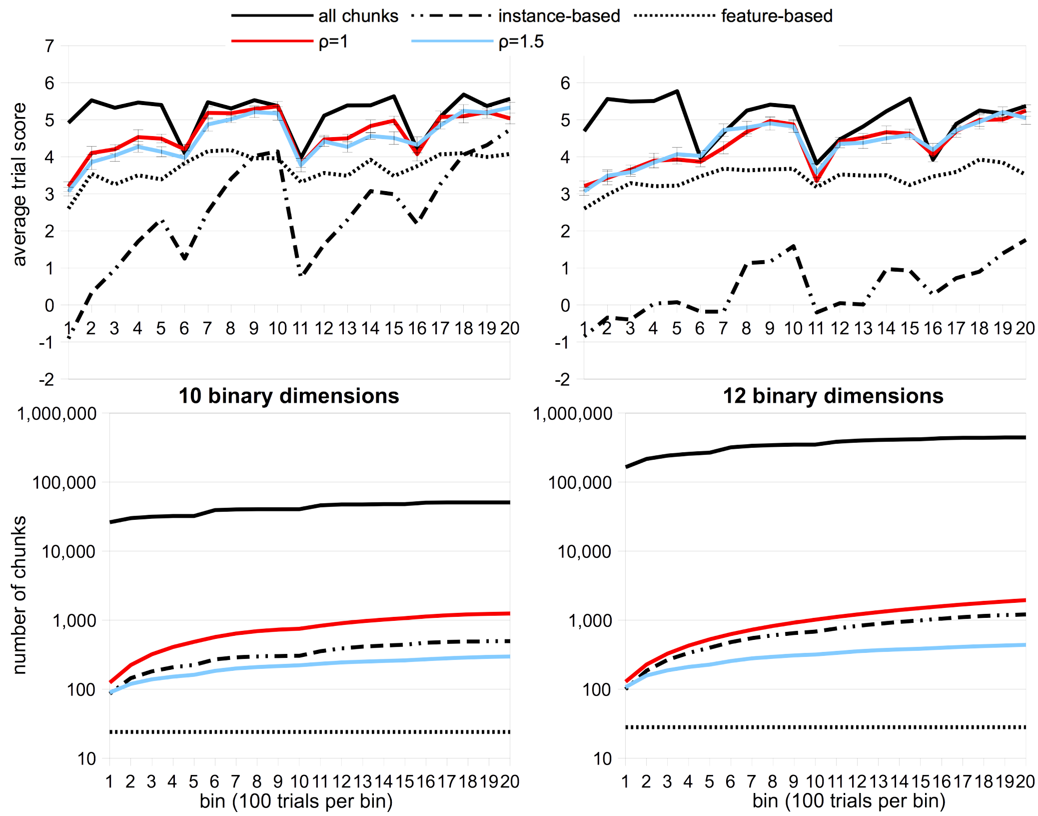


*Figure 3*. Performance and memory growth over time in 10- and 12- binary dimension decision environments. Error bars represent standard error.

-26.5%, and -43.0% in the 10-dimensional environment, and -6.1%, -6.2%, -27.3%, and -86.0% in the 12-dimensional environment, respectively.

Figure 4 displays the relationship between performance and memory consumption in this simulation for both 10- and 12-dimensional environments (y-axis in log-scale). Although it may be obvious that performance can be gained at the expense of memory, what may be less obvious is that (1) small performance gains require exponential chunk growth, and (2) as the environment becomes more complex (from 10 dimensions to 12 dimensions) performance differences become less distinguishable while memory expenses become more distinguishable.

On a final note, the 10- and 12-dimensional binary state decision environments are relatively simple. In more complex environments the all-chunks model is computationally infeasible due to state-space explosion. Additionally, the $\rho = 1$ model produces nearly one chunk per time-step, which makes it computationally infeasible in more persistent environments. The full conservative-rational model ($\rho > 1$) may be the needed alternative to produce high levels of performance while cutting down on memory expenses.

## Summary and Discussion

There is much psychological evidence that states are represented as multiple varying configurations of perceptual cues (e.g. Pearce, 1994). Configural (a.k.a. chunk) representation provides many cognitive advantages, such as extending the size of working memory (e.g. Ericsson et al., 2004), and providing representation for both generic rules and instance exceptions (e.g. Gluck & Bower, 1988b). However, it is computationally implausible to hold all possible cue configu-

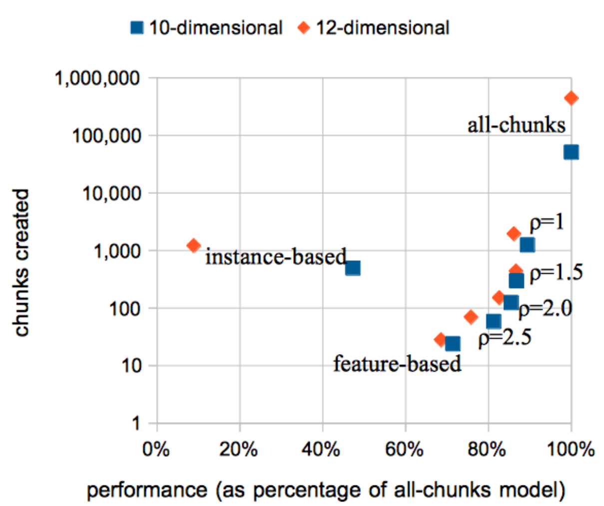


*Figure 4*. Chunk number versus performance in 10- and 12-dimension binary decision environments.

rations in memory for environments of even modest complexity, because the number of chunks grows exponentially as a function of the number of perceptual cues. This is termed the state-space explosion problem.

We propose a model that helps to alleviate state-space explosion via (1) conservative chunk expansion (vis-a-vis *familiarization* in EPAM/CHREST Feigenbaum & Simon, 1984), and (2) rational selection of chunks for expansion (vis-a-vis *base-level activation* in ACT-R; Anderson & Lebiere, 1998). Simulation 1 demonstrates how the proposed model alleviates the state-space explosion, especially in the extremely noisy environments. When chunk generation is based on statistical regularities in the environment (recency/frequency), it aids in distinguishing signal from noise among cue co-occurrences. Simulation 2 presents an analysis of model performance in generic binary decision environments. Results demonstrate a performance-memory tradeoff, where small improvements in performance require exponential sacrifices in memory use. As the environment becomes more complex, small performance improvements require even more memory use.

State-representation, state-space explosion, and memory-size in general are among chief concerns for Artificial Intelligence and Cognitive Architectures (e.g. Bolch, Greiner, de Meer, & Trivedi, 2006; Douglass, Ball, & Rodgers, 2009; Sutton & Barto, 1998; Uther & Veloso, 1998). The current work focuses on limiting memory expansion based on recency/frequency principles, expanding only the memory elements with the highest activation. An alternative approach, proposed by Derbinsky and Laird (2013), employs the recency/frequency principles to delete memory elements with low activation. Future work will focus on employing these principles in concert, to further decrease memory size.

The number of connections needed between chunks, actions, and rewards may be another source of severe memory growth. The time required at each cycle for learning and action-selection processes would increase in proportion with the number of connections in memory. For systems that focus on learning State-Action-Reward transitions and employing these in action-selection (Reinforecement Learning; Sutton & Barto, 1998) the number of stored connections would be a linear function of the number of chunks. For systems that focus on State-Action-State transitions (e.g. Veksler, Gray, & Schoelles, 2013; Voicu & Schmajuk, 2002), or systems that focus on both transition types (e.g. Gläscher, Daw, Dayan, & O'Doherty, 2010; Veksler, Myers, & Gluck, 2014), it would be an exponential function. In addition to minimizing the number of chunks in memory, future work will explore how the principles discussed in this paper may be employed to create connections and to prune them in a more conservative and rational manner.

## Acknowledgements

This research was funded by grant #13RH06COR from the Air Force Office of Scientific Research, and partly conducted while the first author held a National Research Council Research Associateship with the Cognitive Models and Agents branch at the Air Force Research Laboratory.

## References

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum Associates Publishers.

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the Environment in Memory. *Psychological Science*, *2*(6), 396–408.

Bolch, G., Greiner, S., de Meer, H., & Trivedi, K. S. (2006). *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. John Wiley & Sons.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive psychology*, *4*(1), 55–81.

Derbinsky, N., & Laird, J. E. (2013). Effective and Efficient Forgetting of Learned Knowledge in Soar's Working and Procedural Memories. *Cognitive Systems Research*, *24*, 104–113.

Douglass, S., Ball, J., & Rodgers, S. (2009). Large declarative memories in ACT-R. In *9th international conference of cognitive modeling, iccm 2009.*

Ericsson, K. A., Delaney, P. F., Weaver, G., & Mahadevan, R. (2004). Uncovering the structure of a memoristâĂŹs superior âĂIJbasicâĂİ memory capacity. *Cognitive Psychology*, *49*(3), 191–237.

Feigenbaum, E., & Simon, H. (1984). EPAM-like models of recognition and learning. *Cognitive Science*, *8*(4), 305–336.

Gläscher, J., Daw, N., Dayan, P., & O'Doherty, J. (2010). States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning. *Neuron*, *66*(4), 585–595.

Gluck, M. A., & Bower, G. H. (1988a). Evaluating an Adaptive Network Model of Human Learning. *Journal of Memory and Language*, *27*(2), 166–195.

Gluck, M. A., & Bower, G. H. (1988b). From Conditioning to Category Learning - an Adaptive Network Model. *Journal of Experimental Psychology-General*, *117*(3), 227–247.

Gobet, F. (1998). Expert memory: a comparison of four theories. *Cognition*, *66*(2), 115–152.

Gobet, F., Lane, P. C. R., Croker, S., Cheng, P. C. H., Jones, G., Oliver, I., & Pine, J. M. (2001). Chunking mechanisms in human learning. *Trends in cognitive sciences*, *5*(6), 236–243.

Goldstone, R. L. (2000). Unitization during category learning. *Journal of Experimental Psychology: Human Perception and Performance*, *26*(1), 86.

Pearce, J. M. (1994). Similarity and discrimination: a selective review and a connectionist model. *Psychol Review*, *101*(4), 587–607.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: The MIT Press.

Uther, W. T. B., & Veloso, M. M. (1998). Tree based discretization for continuous state space reinforcement learning. In *Aaai/iaai* (pp. 769–774).

Veksler, V. D., Gray, W. D., & Schoelles, M. J. (2007). Categorization and Reinforcement Learning: State Identification in Reinforcement Learning and Network Reinforcement Learning. In *29th annual meeting of the cognitive science society, cogsci07.* Nashville, TN.

Veksler, V. D., Gray, W. D., & Schoelles, M. J. (2013). Goal-Proximity Decision Making. *Cognitive Science*, *37*(4), 757–774. doi: 10.1111/cogs.12034

Veksler, V. D., Myers, C. W., & Gluck, K. A. (2014). SAwSu: An Integrated Model of Associative and Reinforcement Learning. *Cognitive Science*.

Voicu, H., & Schmajuk, N. (2002). Latent learning, shortcuts and detours: a computational model. *Behavioural Processes*, *59*(2), 67–86.

Wagner, A. R., & Rescorla, R. A. (1972). Inhibition in Pavlovian conditioning: Application of a theory. *Inhibition and learning*, 301–336.