An Algorithmic Model of Working Memory Based on Sparse Variational Gaussian Processes

Dongyu Gong* **(dongyu.gong@yale.edu)**
Department of Psychology, Yale University

Mario Belledonne* **(mario.belledonne@yale.edu)**
Department of Psychology, Yale University

Ilker Yildirim (ilker.yildirim@yale.edu)
Department of Psychology & Wu Tsai Institute, Yale University

*These two authors contributed equally to this work.

Abstract

Working memory (WM) involves dynamically manipulating information to support perception, decision-making, and other higher-order cognitive processes. Despite extensive interests in modeling WM, the algorithmic basis of how WM encodes and manipulates information and does so in a goal-driven manner remains unclear. Here, we propose a novel algorithmic model of WM that combines sparse variational Gaussian processes with an adaptive computation algorithm. The model recapitulates a wide range of WM phenomena, including capacity limitations, attraction-repulsion dynamics, and retrocue benefits.

Keywords: working memory; sparse variational Gaussian process; inducing points; adaptive computation

Introduction

Working memory (WM) is a fundamental capability of the human brain that enables the temporary storage and manipulation of information needed for higher-order processes such as decision-making, reasoning, and planning (D'Esposito & Postle, 2015). What are the algorithms by which WM encodes and manipulates information, and does so dynamically in the service of our goals? Despite much progress in modeling WM, including explaining its limited capacity and precision (Luck & Vogel, 1997; Bays, Catalao, & Husain, 2009; Ma, Husain, & Bays, 2014), an algorithmic account of WM encoding and rich maintenance dynamics remains unclear.

Here, we introduce a new computational framework on the algorithmic-level understanding of WM. We model WM representations as sparse Gaussian processes (GPs), where each item/feature is encoded as a probabilistic mapping between perceptual samples and their likelihoods. This framework exposes a quantifiable atomic resource—the inducing points—which form the core of sparse GPs. We leverage these shared inducing points through a recently proposed adaptive computation algorithm (Belledonne, Butkus, Scholl, & Yildirim, in press) that can reallocate them in a goal-directed manner to prioritize task-relevant information in WM.

Model Description

Gaussian Processes for Working Memory

A Gaussian process defines a distribution over functions such that any finite collection of function values has a joint Gaussian distribution: $f(x) \sim \mathcal{GP}(m(x), k(x,x'))$, where $m(x)$ is the mean function and $k(x,x')$ is the covariance kernel. Previous work shows that the human brain can implement GP regression (Friedrich, 2020) and that people use GPs for function learning (Griffiths, Lucas, Williams, & Kalish, 2008). In the context of modeling WM, we can think of GPs as a probabilistic framework that allows us to represent the inherent uncertainty in WM representations.

Sparse GPs with Shared Inducing Points

WM is inherently limited in capacity (Cowan, 2001). In our model, we employ a sparse variational GP framework (Titsias, 2009) that utilizes a shared set of inducing points to represent multiple items in memory (see Figure 1). These inducing points are *targetable* resources that the model can allocate to different items based on task relevance.

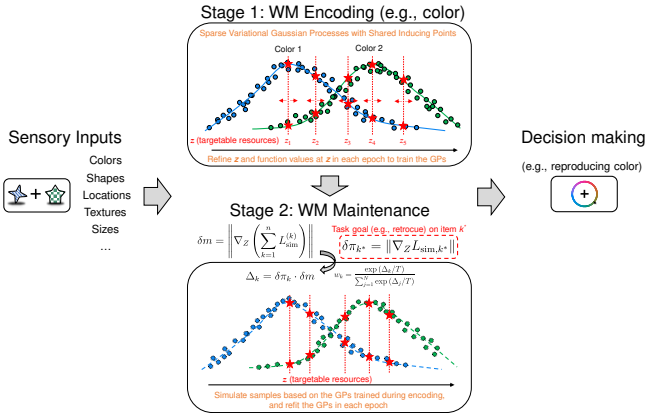

Figure 1: Model schematic. Sensory inputs are represented in sparse variational GPs with shared inducing points. WM maintenance employs an adaptive computation algorithm to optimize decision making on task-relevant items.

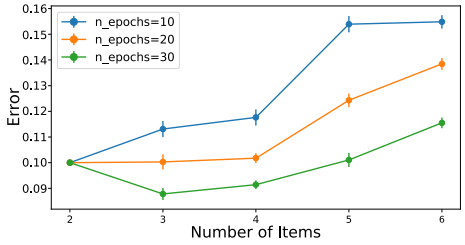

Figure 2: The set size effect. Retrieval error as a function of set size and the number of encoding epochs. Items are in the domain $[0,1)$.

The shared inducing points are defined as a set of R locations in the input space, denoted as $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_R\}$, with corresponding function values $\mathbf{u} = f(\mathbf{Z})$. To implement the encoding process, we train the GP model on perceptual samples and their likelihoods $\{\mathbf{x}_i^{(k)}, \mathbf{y}_i^{(k)}\}_{i=1}^{n}$ for each item k, where n is the number of samples. We approximate the full posterior with a variational distribution $q(\mathbf{u})$. For each item k, the encoding loss is given by:

$$
\begin{aligned}
L_{\text{enc}}^{(k)} &= -\text{ELBO}\left(f_k(\mathbf{x}^{(k)}), \mathbf{y}^{(k)}\right) \\
&= -\mathbb{E}_{q(f^{(k)}(\mathbf{x}))}\left[\log p(\mathbf{y}^{(k)} \mid f^{(k)}(\mathbf{x}))\right] + \text{KL}\left(q(\mathbf{u}) \,\|\, p(\mathbf{u})\right).
\end{aligned}
$$

For a WM system encoding N items, the overall encoding loss is: $L = \frac{1}{N}\sum_{k=1}^{N} L_{\text{enc}}^{(k)}$.

Adaptive Computation Algorithm

After initial encoding, an adaptive computation algorithm dynamically reallocates the limited shared inducing points during WM maintenance based on predicted task goals. In this stage, the WM system no longer has access to the perceptual samples, but instead relies on synthesized samples from the approximated GP posteriors to maintain WM representations.

Synthesizing Samples for Maintenance: For each item k, we sample $\mathbf{x}_{\text{sim}}^{(k)}$ uniformly from the domain and obtain the simulated function value by sampling from the GP predictive distribution: $\mathbf{y}_{\text{sim}}^{(k)} \sim \mathcal{N}\left(f_k^{\text{mean}}(\mathbf{x}_{\text{sim}}^{(k)}), f_k^{\text{var}}(\mathbf{x}_{\text{sim}}^{(k)})\right)$.

Maintenance Loss: For each item k, the maintenance loss based on synthesized samples is: $L_{\text{sim}}^{(k)} = -\text{ELBO}\left(f_k(\mathbf{x}_{\text{sim}}^{(k)}), \mathbf{y}_{\text{sim}}^{(k)}\right)$.

Adaptive Weight Computation: The maintenance losses are reweighted based on two signals, δm and $\delta\pi_k$. **(1)** δm represents the overall change in memory representation, calculated as $\delta m = \left\|\nabla_Z \left(\sum_{k=1}^n L_{\text{sim}}^{(k)}\right)\right\|$. **(2)** $\delta\pi_k$ reflects goal–oriented gradients. For task-relevant item k^*, $\delta\pi_{k^*} = \left\|\nabla_Z L_{\text{sim},k^*}\right\|$, while for task-irrelevant items, $\delta\pi_k = \text{baseline}_\pi$, where baseline_π is a small constant. **(3)** For each item k, compute $\Delta_k = \delta\pi_k \cdot \delta m$, and use a softmax to compute weights: $w_k = \frac{\exp(\Delta_k/T)}{\sum_{j=1}^n \exp(\Delta_j/T)}$. The weighted simulation loss is then $L_{\text{sim}} = \frac{1}{n}\sum_{k=1}^n w_k L_{\text{sim}}^{(k)}$, which is used to update the inducing point locations via gradient descent.

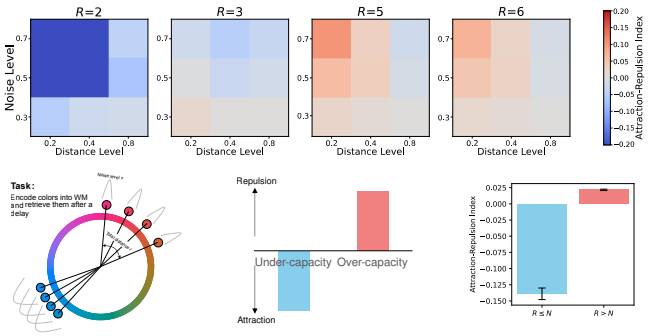

Figure 3: The attraction-repulsion effect. Upper panel: heatmap of the Attraction-Repulsion Index as a function of noise level σ and total distance between items r, with different numbers of shared inducing points (R) representing the same number of items ($N = 4$). Lower Left: task schematic. The model encodes four colors with a varying σ and r. Lower Middle: the hypothetical attraction and repulsion effects. Lower Right: the Attention-Repulsion Index as a function of the relationship between R and N, summarized across the heatmaps.

Experiments

We evaluate our model through a series of simulation experiments designed to capture key WM phenomena.

Set Size Effect

The set size effect, where retrieval error increases with the number of items, arises naturally from the limited resource of WM (Ma et al., 2014). Using a fixed number of inducing points (e.g., $R = 100$) and different numbers of encoding epochs, our simulations confirm that as the number of items stored in the WM system increases, the retrieval error grows (Figure 2), reflecting the limited capacity imposed by the shared inducing points. This aligns well with behavioral findings.

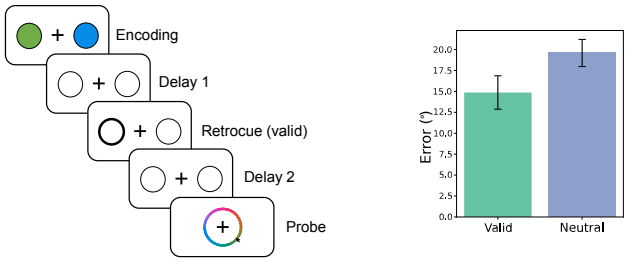

Figure 4: The retrocue effect. Left: a color reproduction task with a retrocue during the delay period. Right: comparison of retrieval error under valid retrocue and neutral conditions.

Attraction-Repulsion Effect

Empirical studies have robustly documented both repulsion and attraction effects as systematic distortions in WM representations (Chunharas, Rademaker, Brady, & Serences, 2022). However, existing computational frameworks have failed to systematically predict when one effect will dominate over the other. To investigate the attraction-repulsion dynamics, we simulate experiments with four WM items while varying the number of shared inducing points, noise level, and distance between items (Figure 3). To quantify biases in memory representations, we define the Attraction-Repulsion Index as $\frac{1}{N}\sum_{i=1}^N \frac{x_i - \mu_c}{r}$, where x_i is the retrieved value for item i, μ_c is the central tendency of the true values, r is the total range of true values, and n is the number of items. Our model captures the transition from attraction (index < 0) in under-capacity conditions to repulsion (index > 0) in over-capacity conditions. These are consistent with the empirical data from Chunharas et al. (2022).

Retrocue Effect

The retrocue effect, wherein cued items are retrieved with higher precision, is a hallmark of the flexibility and goal-drivenness of WM (Griffin & Nobre, 2003; Souza & Oberauer, 2016). As shown in Figure 4, our simulation demonstrates that when a cue directs attention to a specific item, the adaptive computation algorithm reallocates the limited resources effectively and the valid retrocue condition results in significantly lower error, consistent with empirical findings.

Discussion

To provide a fully algorithmic account of WM, we have to be concrete about the resources that realize WM processes (i.e., the inducing points in our sparse GP) and the consequence of exerting this resource on WM representations and the downstream decision-making. Adaptive computation rations these resources using the simple product of these two consequences. Through a series of simulation experiments, we show that our model can provide mechanistic insights into how WM encodes and flexibly maintains information. Future work can extend this model to explore interactions between WM and other cognitive systems (e.g., extending the synthesis and loss to involve priors based on long-term memory) and further validate the proposed computational mechanisms.

References

Bays, P. M., Catalao, R. F. G., & Husain, M. (2009). The precision of visual working memory is set by allocation of a shared resource. *Journal of Vision*, *9*(10), 7–7.

Belledonne, M., Butkus, E., Scholl, B. J., & Yildirim, I. (in press). Adaptive computation as a new mechanism for human attention. *Psychological Review*.

Chunharas, C., Rademaker, R. L., Brady, T. F., & Serences, J. T. (2022). An adaptive perspective on visual working memory distortions. *Journal of Experimental Psychology: General*, *151*(10), 2300.

Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and brain sciences*, *24*(1), 87–114.

D'Esposito, M., & Postle, B. (2015). The cognitive neuroscience of working memory. *Annual Review of Psychology*, *66*(1), 115–142.

Friedrich, J. (2020). Neuronal gaussian process regression. *Advances in Neural Information Processing Systems*, *33*, 7090–7100.

Griffin, I., & Nobre, A. (2003). Orienting attention to locations in internal representations. *Journal of Cognitive Neuroscience*, *15*(8), 1176–1194.

Griffiths, T., Lucas, C., Williams, J., & Kalish, M. (2008). Modeling human function learning with gaussian processes. *Advances in neural information processing systems*, *21*.

Luck, S. J., & Vogel, E. K. (1997). The capacity of visual working memory for features and conjunctions. *Nature*, *390*(6657), 279–281.

Ma, W. J., Husain, M., & Bays, P. M. (2014). Changing concepts of working memory. *Nature Neuroscience*, *17*(3), 347–356.

Souza, A., & Oberauer, K. (2016). In search of the focus of attention in working memory: 13 years of the retro-cue effect. *Attention, Perception, & Psychophysics*, *78*(7), 1839–1860.

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics* (pp. 567–574).