



## Exploring trade-offs between sequential and parallel computation in recurrent neural networks

Marcel Graetz<sup>1</sup>, Caroline Haimerl<sup>1</sup>, Daniel McNamee<sup>\* 1</sup>, Alfonso Renart<sup>\* 1</sup>

<sup>1</sup> Champalimaud Foundation, Lisbon, PT

<sup>\*</sup> equal contribution

Parallel and serial processing modes have been studied widely in humans and other animals [1]. While serial processing is thought to enable few-shot generalization and rule-based reasoning [1,2], extensive practice of specific skills yields more parallel, habit-like computation, increasing speed by reducing the number of reasoning steps [3], but likely at the cost of increased representational resources [4].

Here we explore the trade-off between processing time and representational resources in a recurrent neural network model. Starting from a simple feedforward auto-encoder, we add a recurrent decoder layer and a feedback connection from this layer to the encoder. When the bottleneck is narrower than the intrinsic dimensionality of the data, this recurrence can be used to iteratively “gate” extra information, improving reconstruction relative to the feedforward architecture at the cost of additional time steps.

We train the model end-to-end to reconstruct static noise vectors with a mean-squared error loss and backpropagation through time. The bottleneck is either continuous or discrete; in the latter case we use the Gumbel-softmax reparametrization trick.

In both cases the network finds near-optimal latent codes, e.g. compressing the inputs into a sequential binary code to traverse a binary bottleneck. More generally, systematically varying the bottleneck width forces either sequential, mixed or parallel processing.

Furthermore, preliminary continual-learning experiments suggest that the code reorganizes when the input distribution shifts: the network continually adopts the encoding that maximizes information transmission rate, mirroring adaptive habit formation [3].

Future work will scale the framework to richer datasets, test whether the learned dynamics support compositional generalisation, and derive quantitative predictions for neural data.

**sequential/parallel processing, recurrent neural networks, bottlenecks, autoencoders, habit formation**