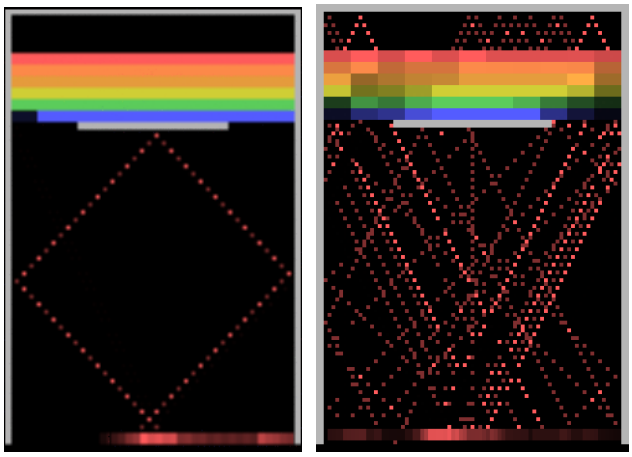
# Schema Networks: Zero-shot Transfer with a Generative Causal Model of Intuitive Physics (Supplementary)

**Ken Kansky  Tom Silver  David A. Mély  Mohamed Eldawy  Miguel Lázaro-Gredilla  Xinghua Lou**
**Nimrod Dorfman  Szymon Sidor  Scott Phoenix  Dileep George**

## 1. Breakout playing visualizations

See https://vimeopro.com/user45297729/schema-networks for visualizations of Schema Networks playing different variations of Breakout after training only on basic Breakout.

Figure 1 shows typical gameplay for one variation.



(a) A3C (b) Schema Networks

*Figure 1.* Screen average during the course of gameplay in the Midwall variation when using A3C and Schema Networks. SNs are able to purposefully avoid the middle wall most of the time, whereas A3C struggles to score any points.

## 2. LP-based Greedy Schema Learning

The details of the LP-based learning of Section 4.2 are provided here.

All authors affiliated with Vicarious AI, California, USA. Correspondence to: Ken Kansky <ken@vicarious.com>, Tom Silver <tom@vicarious.com>.

*Proceedings of the $34^{th}$ International Conference on Machine Learning*, Sydney, Australia, PMLR 70, 2017. 

**Algorithm 1** LP-based greedy schema learning

**Input:** Input vectors $\{x_n\}$ for which $f_W(x_n) = 0$ (the current schema network predicts 0), and the corresponding output scalars $y_n$

1: **Find a cluster of input samples** that can be solved with a single (relaxed) schema while keeping perfect precision (no false alarms). Select an input sample and put it in the set "solved", then solve the LP

$$\begin{aligned} \min_{w \in [0,1]^D} & \sum_{n:y_n=1} (1-x_n)w \\ \text{s.t. } & (1-x_n)w > 1 \quad \forall_{n:y_n=0} \\ & (1-x_n)w = 0 \quad \forall_{n \in \text{solved}} \end{aligned}$$

2: **Simplify the resulting schema**. Put all the input samples for which $(1-x_n)w = 0$ in the set "solved". Simplify the just found schema $w$ by making it as sparse as possible while keeping the same precision and recall:

$$\begin{aligned} \min_{w \in [0,1]^D} & w^T \vec{1} \\ \text{s.t. } & (1-x_n)w > 1 \quad \forall_{n:y_n=0} \\ & (1-x_n)w = 0 \quad \forall_{n \in \text{solved}} \end{aligned}$$

3: **Binarize the schema**. In practice, the found $w$ is binary most of the time. If it is not, repeat the previous minimization using binary programming, but optimize only over the elements of $w$ that were found to be non-zero. Keep the rest clamped to zero.

**Output:** New schema $w$ to add to the network