

# Powerset Convolutional Neural Networks

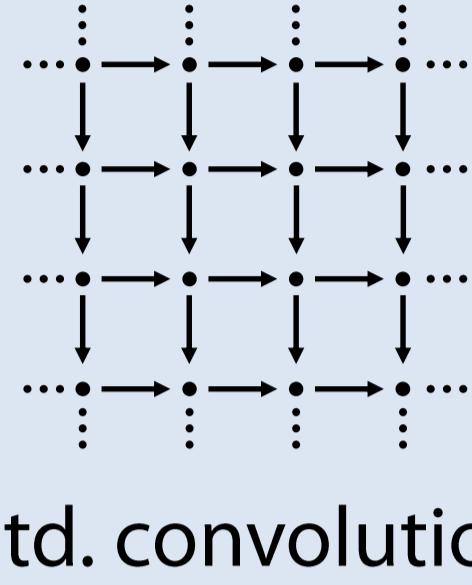
Chris Wendler, Computer Science, ETH Zurich

Dan Alistarh, IST Austria

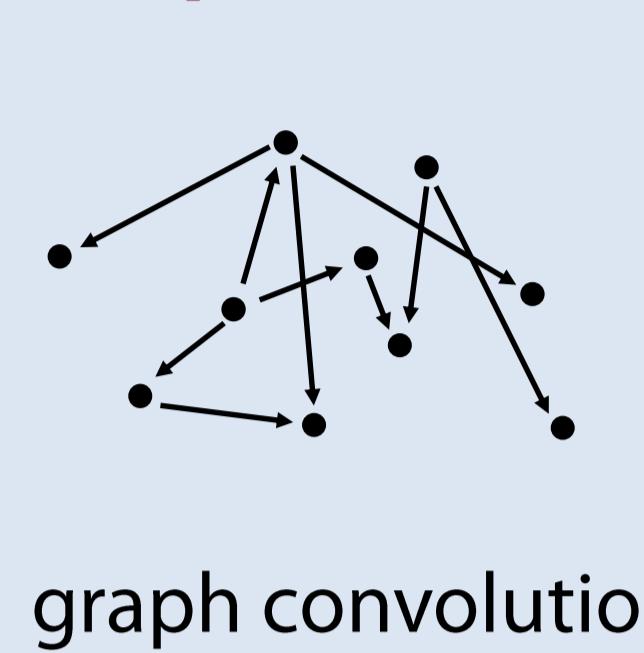
Markus Püschel, Computer Science, ETH Zurich

## Goal

### Classical CNN



### Graph CNN



### New Powerset CNN



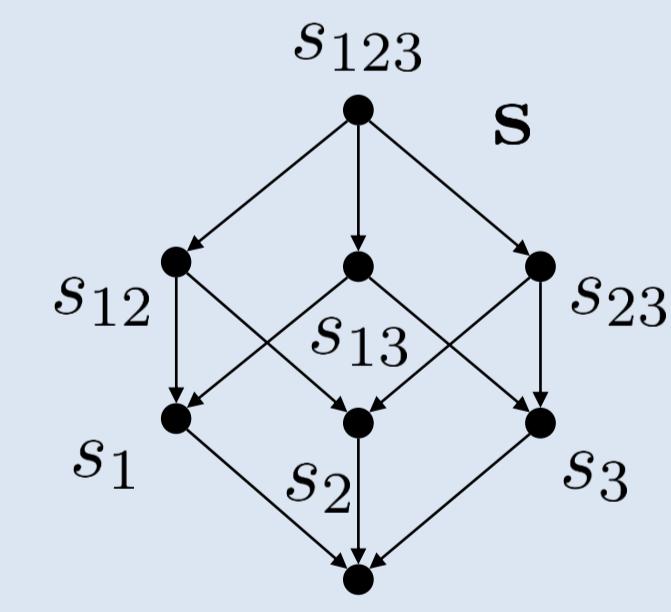
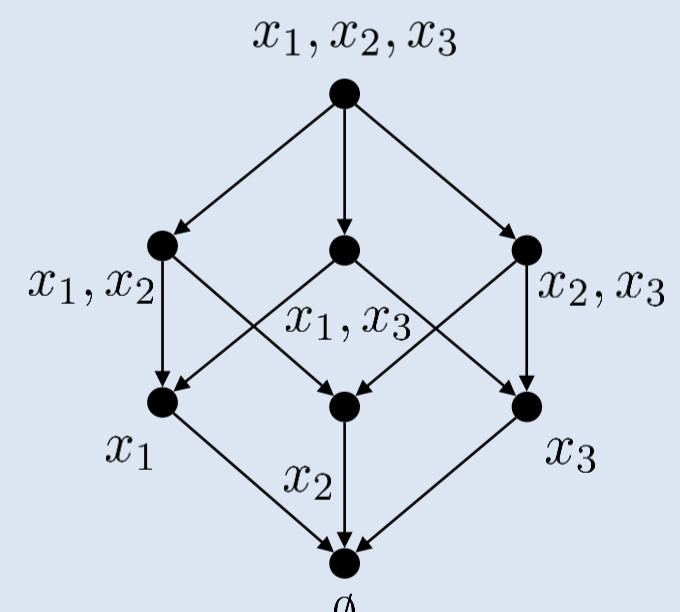
### Derivation: Algebraic Signal Processing (ASP)

shift  $\longrightarrow$  shift equivariant convolution  $\longrightarrow$  neural layers etc.

## Powerset Convolution

Consider the ground set  $N = \{x_1, \dots, x_n\}$ .

### Powerset = Hypercube



### Signal (= set function)

$$\mathbf{s} = (s_A)_{A \subseteq N} \in \mathbb{R}^{2^n}$$

### Shifted Signal

for all  $Q \subseteq N$

$$T_Q \mathbf{s} = (s_{A \setminus Q})_{A \subseteq N}$$

### Filter (= set function)

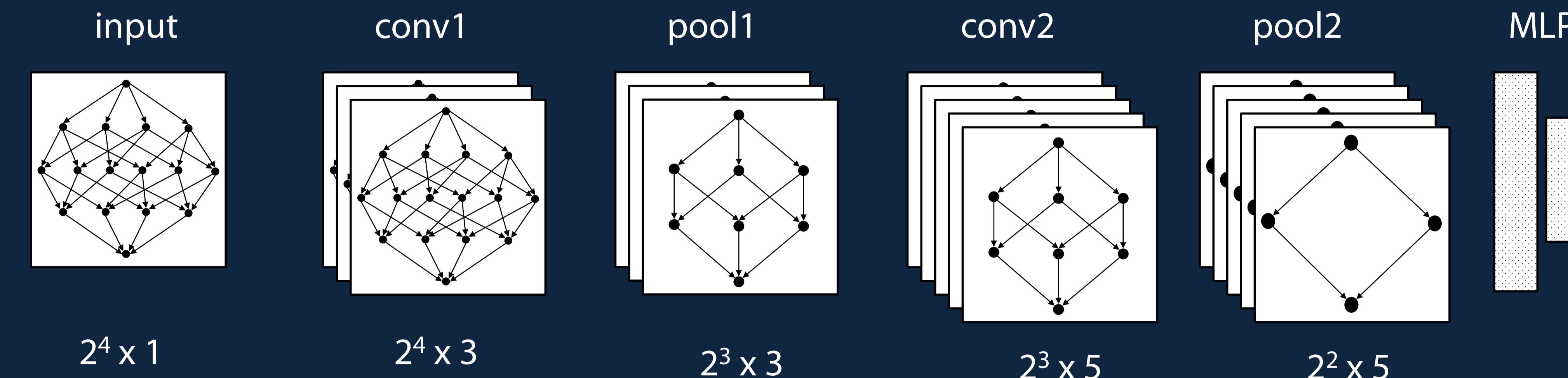
$$\mathbf{h} = (h_Q)_{Q \subseteq N} \in \mathbb{R}^{2^n}$$

### Convolution

$$(\mathbf{h} * \mathbf{s})_A = \sum_{Q \subseteq N} h_Q s_{A \setminus Q}$$

### Equivariance

$$\mathbf{h} * T_Q \mathbf{s} = T_Q (\mathbf{h} * \mathbf{s})$$



## Pattern Matching Perspective

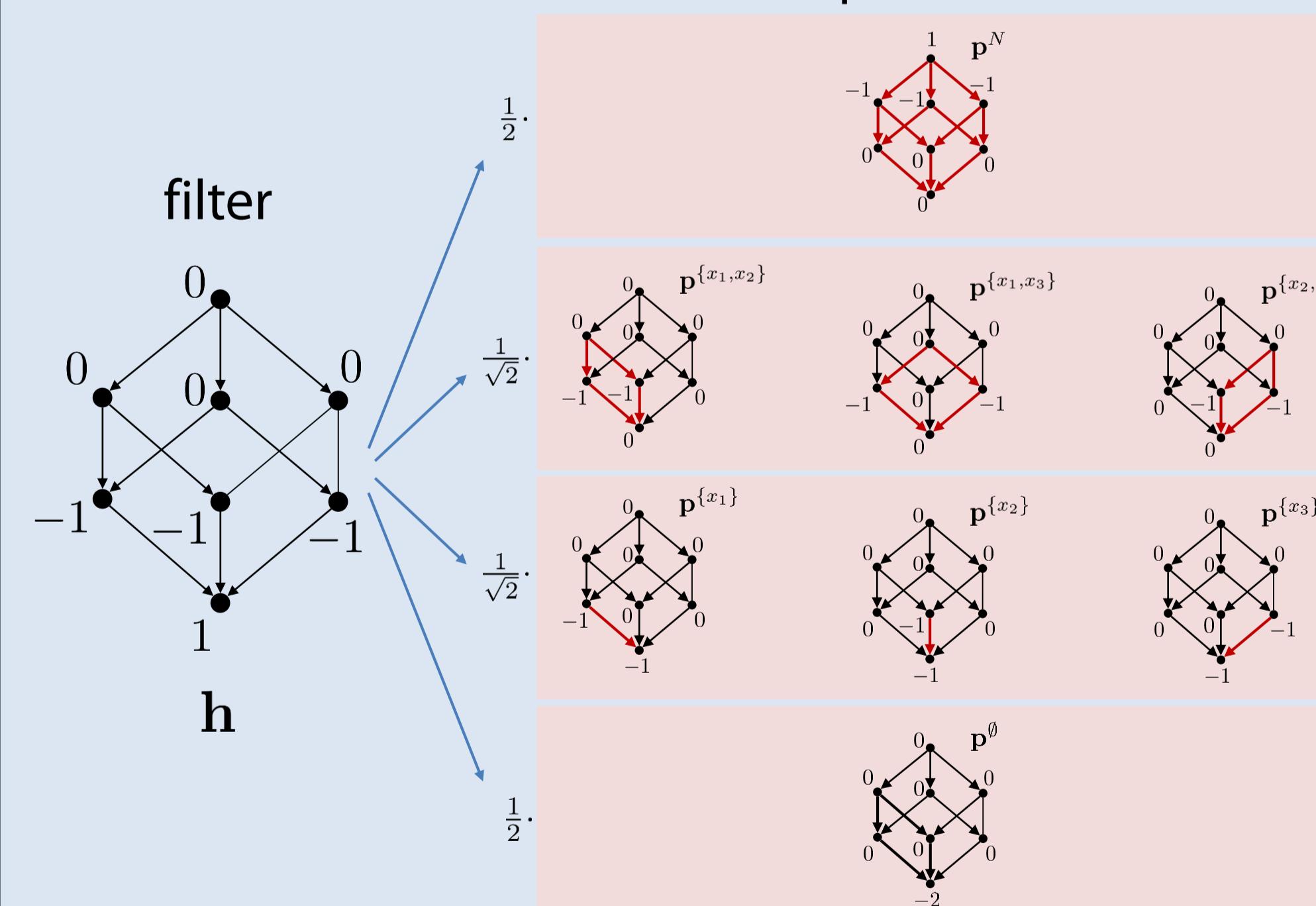
Let  $\mathbf{h} \in \mathbb{R}^{2^n}$  be a filter.

$$\text{Pattern} \quad \mathbf{p}^A := \arg \max_{\mathbf{s} \in \mathbb{R}^{2^n}: \|s\|=1} (\mathbf{h} * \mathbf{s})_A$$

**Images**  $\mathbf{h} = \begin{bmatrix} -1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix}$   $\mathbf{p}^{(i,j)} = \dots \begin{bmatrix} -1 & 1 & -1 \\ 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix} \dots$  at any location

$$(\mathbf{h} * \mathbf{s})_{i,j} = \sum_{k,l \in \mathbb{Z}} h_{k,l} s_{i-k, j-l} = \sum_{k,l \in \mathbb{Z}} h_{i+k, j+l} s_{k,l}$$

### Powerset



### General Equation

$$\sum_{Q \subseteq N} h_Q s_{A \setminus Q} = \sum_{Q_1 \subseteq A} \left( \underbrace{\sum_{Q_2 \subseteq N \setminus A} h_{Q_1 \cup Q_2}}_{=: h'_{Q_1}} \right) s_{A \setminus Q_1}$$

$$\mathbf{p}_B^A = \frac{1}{\|\mathbf{h}'\|} \begin{cases} h'_{A \setminus B} & \text{if } B \subseteq A \\ 0 & \text{otherwise.} \end{cases}$$

## Powerset CNN

### Convolutional Layer

$n_c$  input-,  $n_f$  output channels, and  $\sigma(x) = \max(0, x)$ :

$$t_A^{(j)} = \sigma(\sum_{i=1}^{n_c} (h^{(i,j)} * s^{(i)})_A)$$

### Pooling Layer

$n_c$  input-,  $n_c$  output channels:

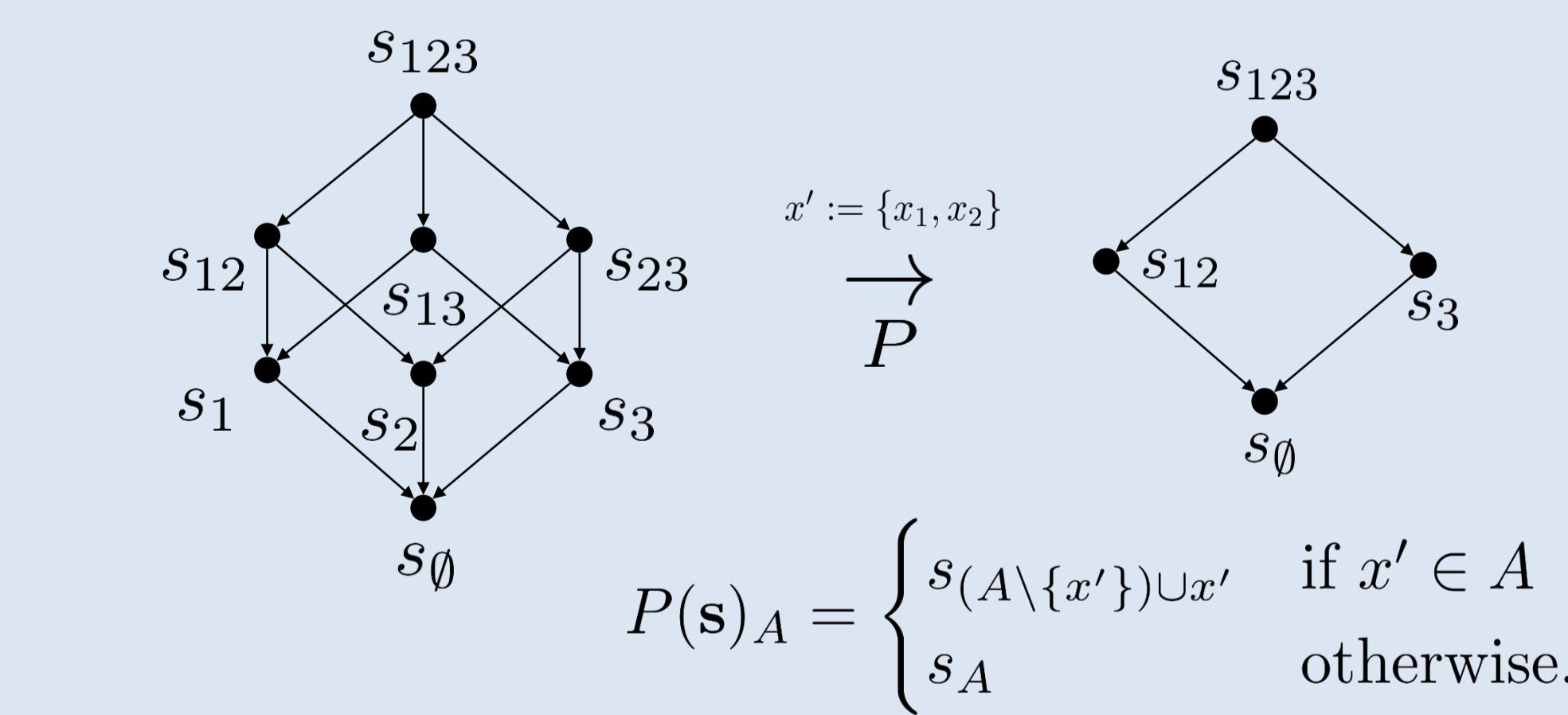
$$t^{(i)} = P(s^{(i)})$$

e.g., combine elements in  $N$  or combine elements in  $2^N$

### Groundset Pooling

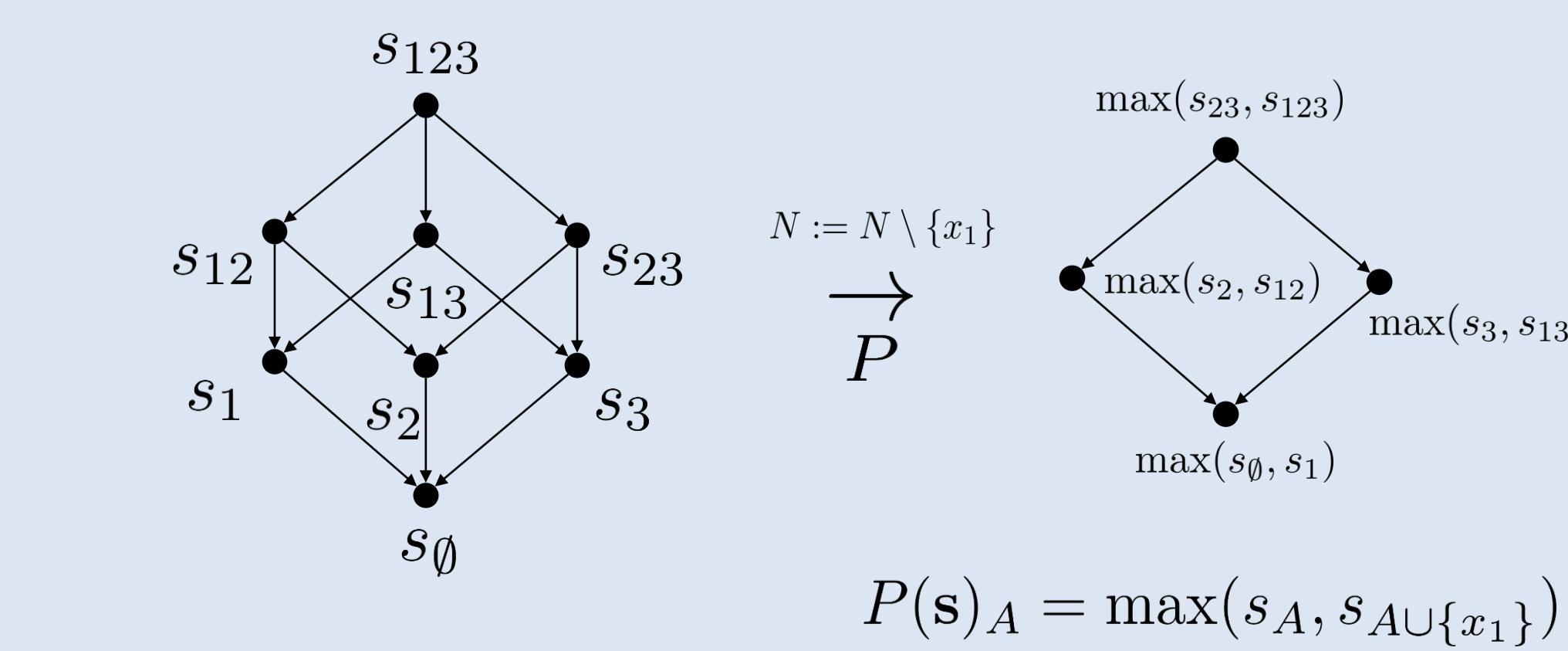
Idea: combine two elements

$$x' := \{x_1, x_2\} \text{ and } N' := \{x', x_3, \dots, x_n\}$$



### Powerset Pooling

Idea: remove an element  $N' := N \setminus \{x_1\}$



## Experimental Evaluation

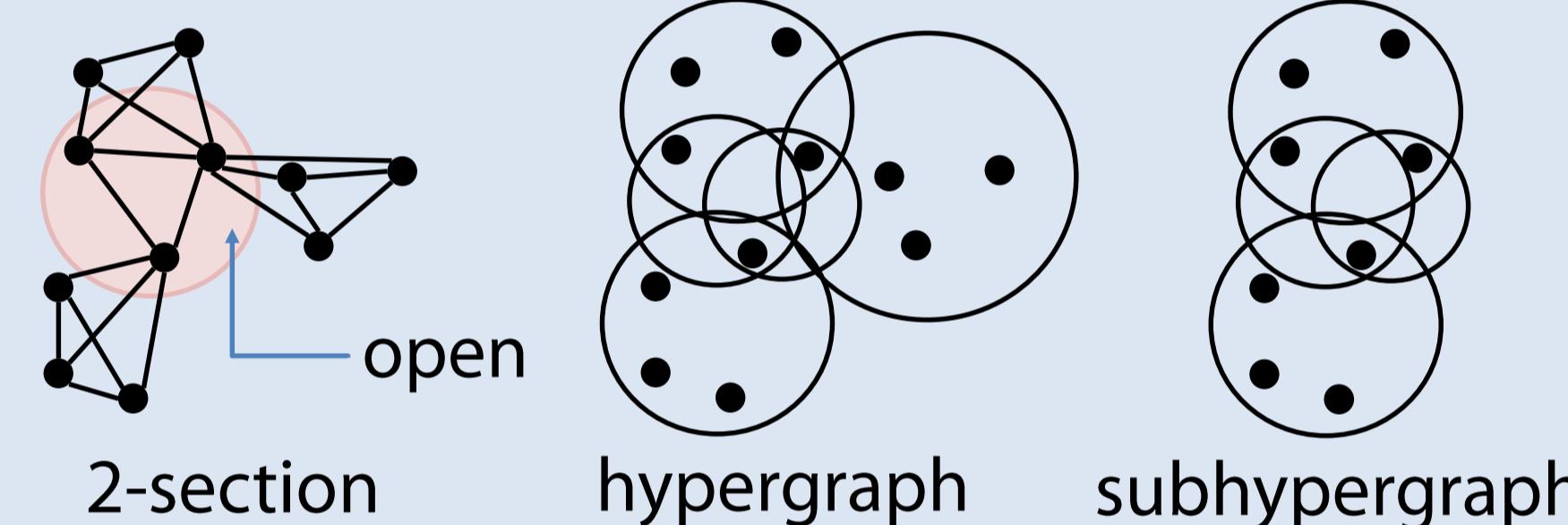
**Spectral Patterns** sample spectral coefficients with class-dependent support in powerset Fourier domain.  
→ given such a signal, what is the generating class?

**k-Junta** sample set functions that only depend on  $k$  of the  $n$  elements of the ground set. → given a  $k$ -junta, what is  $k$ ? There is  $N' \subset N$  with  $|N'| = k$ :  $s(A) = s(A \cap N')$ .

**Submodularity** sample submodular and almost submodular set functions → given a set function, is it submodular? For  $A \subseteq B$ :  $s_A - s_A \geq s_B - s_B$ .

**Hyperedge Closure** a set of vertices is called open if it is a clique in the 2-section of the hypergraph and not contained in a hyperedge. → given a subhypergraph, is it open or closed? (COAUTH10, CON10)

**Source Hypergraph** sample subhypergraphs from different source hypergraphs. → given a hypergraph, what is the source hypergraph? (DOM4, DOM6)



## Results

	Patterns	$k$ -Junta	Submod.	COAUTH10	CON10	DOM4	DOM6
Baselines							
MLP	$46.8 \pm 3.9$	$43.2 \pm 2.5$	-	$80.7 \pm 0.2$	$66.1 \pm 1.8$	$93.6 \pm 0.2$	$71.1 \pm 0.3$
L-GCN	$52.5 \pm 0.9$	$69.3 \pm 2.8$	-	$84.7 \pm 0.9$	$67.2 \pm 1.8$	$96.0 \pm 0.2$	$73.7 \pm 0.4$
L-GCN pool	$45.0 \pm 1.0$	$60.9 \pm 1.5$	-	$83.2 \pm 0.7$	$65.7 \pm 1.0$	$93.2 \pm 1.1$	$71.7 \pm 0.5$
L-GCN pool avg.	$42.1 \pm 0.3$	$64.3 \pm 2.2$	$82.2 \pm 0.4$	$56.8 \pm 1.1$	$64.1 \pm 1.7$	$88.4 \pm 0.3$	$62.8 \pm 0.4$
A-GCN	$65.5 \pm 0.9$	$95.8 \pm 2.7$	-	$80.5 \pm 0.7$	$64.9 \pm 1.8$	$93.9 \pm 0.3$	$69.1 \pm 0.5$
A-GCN pool	$56.9 \pm 2.2$	$91.9 \pm 2.1$	$89.8 \pm 1.8$	$84.1 \pm 0.6$	$66.0 \pm 1.6$	$93.8 \pm 0.3$	$70.7 \pm 0.4$
A-GCN pool avg.	$54.8 \pm 0.9$	$95.8 \pm 1.1$	$84.8 \pm 1.9$	$64.8 \pm 1.1$	$65.4 \pm 0.7$	$92.7 \pm 0.6$	$67.9 \pm 0.3$
Proposed models							
*-PCN	$88.5 \pm 4.3$	$97.2 \pm 2.3$	$88.6 \pm 0.4$	$80.6 \pm 0.7$	$62.8 \pm 2.9$	$94.1 \pm 0.3$	$70.5 \pm 0.3$
*-PCN pool	$80.9 \pm 0.9$	$96.0 \pm 1.6$	$85.1 \pm 1.8$	$82.6 \pm 0.4$	$62.9 \pm 2.0$	$94.0 \pm 0.3$	$70.2 \pm 0.5$
*-PCN pool avg.	$75.9 \pm 1.9$	$96.5 \pm 0.6$	$87.0 \pm 1.6$	$80.6 \pm 0.5$	$63.4 \pm 3.5$	$94.4 \pm 0.3$	$73.0 \pm 0.3$
○-PCN	-	$97.5 \pm 1.4$	-	$83.6 \pm 0.4$	$68.7 \pm 1.3$	$93.7 \pm 0.2$	$69.9 \pm 0.3$
○-PCN pool	-	$96.4 \pm 1.7$	-	$84.8 \pm 0.3$	$68.2 \pm 0.8$	$93.6 \pm 0.3$	$70.3 \pm 0.4$
○-PCN pool avg.	$54.8 \pm 1.9$	$96.6 \pm 0.7$	$80.9 \pm 2.9$	$83.3 \pm 0.5$	$67.0 \pm 2.0$	$94.8 \pm 0.3$	$73.5 \pm 0.5$

## Algebraic Signal Processing

<https://www.ece.cmu.edu/~smart/> <https://acl.inf.ethz.ch/research/ASP/>

**Framework to generalize standard DSP**  
**ASP is constructive:** derived from shift

