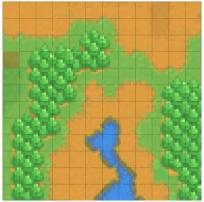
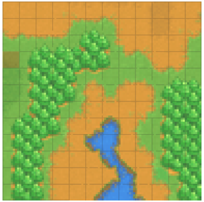
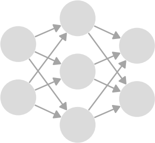
EXPLAIN, AGREE, LEARN:
Scaling Learning for Neural Probabilistic Logic

Victor Verreet, **Lennert De Smet**, Luc De Raedt and Emanuele Sansone

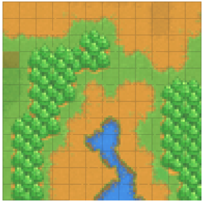


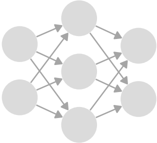




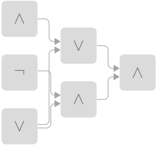
$$\boldsymbol{x} \in \{\texttt{grass}, \texttt{sand}, \texttt{water}, \texttt{tree}, \texttt{mountain}\}^{12\times 12}$$

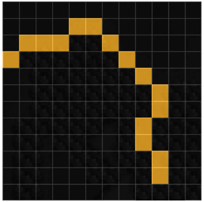
$$p_\Lambda(\boldsymbol{x})$$





$$\boldsymbol{x} \in \{\texttt{grass}, \texttt{sand}, \texttt{water}, \texttt{tree}, \texttt{mountain}\}^{12\times 12}$$

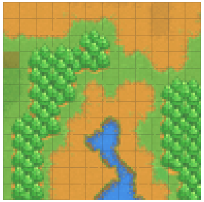


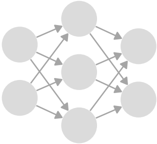


$p_\wedge(\boldsymbol{x})$

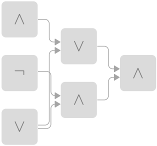
$\varphi \leftrightarrow \texttt{dijkstra}(\boldsymbol{x}) \texttt{ is shortest}$

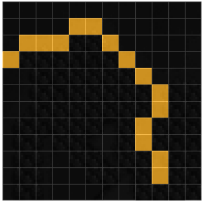
$\mathsf{WMC}(\varphi) = \mathbb{P}(\varphi)$ (intractable!)





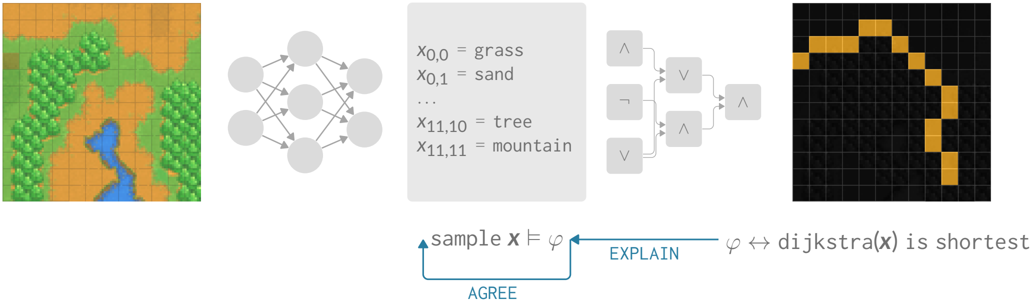
$x_{0,0}$ = grass
$x_{0,1}$ = sand
...
$x_{11,10}$ = tree
$x_{11,11}$ = mountain





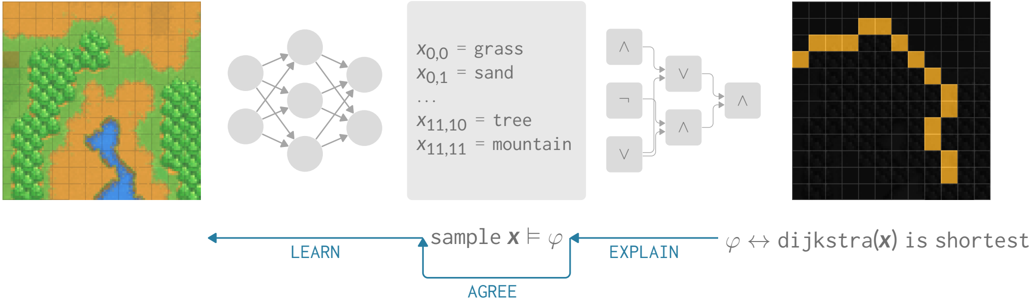
sample $\boldsymbol{x} \models \varphi$ $\xleftarrow{\text{EXPLAIN}}$ $\varphi \leftrightarrow$ dijkstra($\boldsymbol{x}$) is shortest

EXPLAIN Sample **some** explanations for φ



EXPLAIN Sample **some** explanations for φ

AGREE Align samples with neural predictions



EXPLAIN	Sample **some** explanations for φ
AGREE	Align samples with neural predictions
LEARN	Learn from the aligned samples via variational objective

Learning through all logical solutions does not scale but is also not necessary

Setting Neurosymbolic AI informs neural networks
by supervising with logical solutions

Learning through all logical solutions does not scale but is also not necessary

Setting Neurosymbolic AI informs neural networks
by supervising with logical solutions

Learning through all logical solutions does not scale but is also not necessary

Setting Neurosymbolic AI informs neural networks
by supervising with logical solutions

Problem Getting all logical solutions is #P-hard
and bottlenecks neurosymbolic learning

Learning through all logical solutions does not scale but is also not necessary

Setting Neurosymbolic AI informs neural networks
by supervising with logical solutions

Problem Getting all logical solutions is #P-hard
and bottlenecks neurosymbolic learning

Our solution Sampled logical solutions form a variational objective
as a surrogate that scales better and provides guarantees

1. Explanations induce a variational surrogate objective
2. A generic algorithm to sample explanations with bounds
3. Verification of bounds and improved scaling

A variational objective supported on explanations

$$\log \mathbb{P}(\varphi) =$$

A variational objective supported on explanations

$$\log \mathbb{P}(\varphi) = \log \sum_{\boldsymbol{x} \models \varphi} p_{\Lambda}(\boldsymbol{x})$$

A variational objective supported on explanations

$$\log \mathbb{P}(\varphi) = \log \sum_{\boldsymbol{x} \models \varphi} p_\Lambda(\boldsymbol{x}) = \log \sum_{\boldsymbol{x} \models \varphi} \frac{q(\boldsymbol{x}) p_\Lambda(\boldsymbol{x})}{q(\boldsymbol{x})}$$

A variational objective supported on explanations

$$\log \mathbb{P}(\varphi) = \log \sum_{\boldsymbol{x} \models \varphi} p_\Lambda(\boldsymbol{x}) = \log \sum_{\boldsymbol{x} \models \varphi} \frac{q(\boldsymbol{x}) p_\Lambda(\boldsymbol{x})}{q(\boldsymbol{x})}$$

$$\geq \sum_{\boldsymbol{x} \models \varphi} q(\boldsymbol{x}) \log \frac{p_\Lambda(\boldsymbol{x})}{q(\boldsymbol{x})}$$

A variational objective supported on explanations

$$\log \mathbb{P}(\varphi) = \log \sum_{\boldsymbol{x} \models \varphi} p_\Lambda(\boldsymbol{x}) = \log \sum_{\boldsymbol{x} \models \varphi} \frac{q(\boldsymbol{x}) p_\Lambda(\boldsymbol{x})}{q(\boldsymbol{x})}$$

$$\geq \sum_{\boldsymbol{x} \models \varphi} q(\boldsymbol{x}) \log \frac{p_\Lambda(\boldsymbol{x})}{q(\boldsymbol{x})}$$

Upshot Any proposal distribution $\boldsymbol{q}$ over **some solutions** of φ yields a valid lower bound on $\log \mathbb{P}(\varphi)$

The optimal proposal agrees with the neural network

The optimal proposal $\boldsymbol{q}$ on **explanations** $\mathcal{E}_\varphi$ chooses

$$q(\boldsymbol{e}) = \frac{\boxed{p_\Lambda(\boldsymbol{e})}}{\sum_{\boldsymbol{e}' \in \mathcal{E}_\varphi} p_\Lambda(\boldsymbol{e}')}$$

The optimal proposal agrees with the neural network

The optimal proposal $\boldsymbol{q}$ on **explanations** $\mathcal{E}_\varphi$ chooses

$$q(\boldsymbol{e}) = \frac{p_\Lambda(\boldsymbol{e})}{\sum_{\boldsymbol{e}' \in \mathcal{E}_\varphi} p_\Lambda(\boldsymbol{e}')}$$

Upshot The best $\boldsymbol{q}$ weighs explanations like the neural network

Learning from a set of explanations instead of all solutions

EXPLAIN Get a set $\mathcal{E}_\varphi$ of explanations for φ

Learning from a set of explanations instead of all solutions

EXPLAIN — Get a set $\mathcal{E}_\varphi$ of explanations for φ

AGREE — Proposal distribution $\boldsymbol{q}$ is mixture of explanations

$$q(\boldsymbol{x}) = \frac{\mathbb{1}_{\boldsymbol{x}\in\mathcal{E}_\varphi} p_\Lambda(\boldsymbol{x})}{\sum_{\boldsymbol{e}\in\mathcal{E}_\varphi} p_\Lambda(\boldsymbol{e})}$$

Learning from a set of explanations instead of all solutions

EXPLAIN Get a set $\mathcal{E}_\varphi$ of explanations for φ

AGREE Proposal distribution $\boldsymbol{q}$ is mixture of explanations

$$q(\boldsymbol{x}) = \frac{\mathbb{1}_{\boldsymbol{x} \in \mathcal{E}_\varphi} p_\Lambda(\boldsymbol{x})}{\sum_{\boldsymbol{e} \in \mathcal{E}_\varphi} p_\Lambda(\boldsymbol{e})}$$

LEARN Maximize the variational lower bound

$$\log \sum_{\boldsymbol{e} \in \mathcal{E}_\varphi} p_\Lambda(\boldsymbol{e})$$

1. Explanations induce a variational surrogate objective
2. A generic algorithm to sample explanations and bounds
3. Verification of bounds and improved scaling

DPLL as a generic explanation sampler

Sample next assignment $\boldsymbol{x}'$
instead of all assignments

Input: formula φ, assignment $\boldsymbol{x}$, sampling strategy $\boldsymbol{s}$
Output: explanation for φ

$\boldsymbol{x} \leftarrow$ propagate$(\varphi, \boldsymbol{x})$ ▷ trivial assignments
if $\boldsymbol{x}$ is an explanation for φ **then**
  return $\boldsymbol{x}$ ▷ explanation found
else if $\boldsymbol{x}$ cannot satisfy φ **then**
  return EXPLAIN$(\varphi, \boldsymbol{x}_?, s)$ ▷ conflict restart
end if
$\boldsymbol{x}' \leftarrow$ sample from $\boldsymbol{s}(\boldsymbol{x})$ ▷ new assignment
return EXPLAIN$(\varphi, \boldsymbol{x}', s)$ ▷ recurse

DPLL as a generic explanation sampler

Sample next assignment $\boldsymbol{x}'$
instead of all assignments

Sampling strategy s is free
and can be parametrised

Input: formula φ, assignment $\boldsymbol{x}$, sampling strategy s
Output: explanation for φ

$\boldsymbol{x} \leftarrow \text{propagate}(\varphi, \boldsymbol{x})$ ▷ trivial assignments
if $\boldsymbol{x}$ is an explanation for φ **then**
    return $\boldsymbol{x}$ ▷ explanation found
else if $\boldsymbol{x}$ cannot satisfy φ **then**
    return EXPLAIN$(\varphi, \boldsymbol{x}_?, s)$ ▷ conflict restart
end if
$\boldsymbol{x}' \leftarrow$ sample from $s(\boldsymbol{x})$ ▷ new assignment
return EXPLAIN$(\varphi, \boldsymbol{x}', s)$ ▷ recurse

Every set of explanations provides an upper and lower bound and diversity improves bounds

Summing probabilities of $\mathcal{E}_\varphi$
lower bounds the WMC

Every set of explanations provides an upper and lower bound and diversity improves bounds

Summing probabilities of $\mathcal{E}_\varphi$
lower bounds the WMC

$$\tilde{\mathbb{P}}(\varphi) = \sum_{\boldsymbol{e} \in \mathcal{E}_\varphi} p_\Lambda(\boldsymbol{e})$$

Every set of explanations provides an upper and lower bound and diversity improves bounds

Summing probabilities of $\mathcal{E}_\varphi$
lower bounds the WMC

$$\tilde{\mathbb{P}}(\varphi) = \sum_{\boldsymbol{e} \in \mathcal{E}_\varphi} p_\Lambda(\boldsymbol{e}) \leq \sum_{\boldsymbol{x} \models \varphi} p_\Lambda(\boldsymbol{x})$$

Every set of explanations provides an upper and lower bound and diversity improves bounds

Summing probabilities of $\mathcal{E}_\varphi$
lower bounds the WMC

$$\tilde{\mathbb{P}}(\varphi) = \sum_{\boldsymbol{e} \in \mathcal{E}_\varphi} p_\Lambda(\boldsymbol{e}) \leq \sum_{\boldsymbol{x} \models \varphi} p_\Lambda(\boldsymbol{x}) = \mathbb{P}(\varphi) = \mathtt{WMC}(\varphi)$$

Every set of explanations provides an upper and lower bound and diversity improves bounds

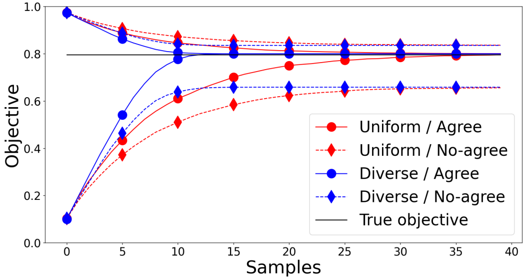
Summing probabilities of $\mathcal{E}_\varphi$
lower bounds the WMC

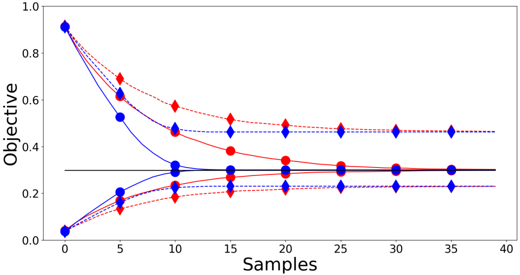
$$\tilde{\mathbb{P}}(\varphi) = \sum_{\boldsymbol{e} \in \mathcal{E}_\varphi} p_\Lambda(\boldsymbol{e}) \leq \sum_{\boldsymbol{x} \models \varphi} p_\Lambda(\boldsymbol{x}) = \mathbb{P}(\varphi) = \mathrm{WMC}(\varphi)$$

Implication A sampler with a **diverse** sampling strategy
will provide a better lower bound through a larger $\mathcal{E}_\varphi$

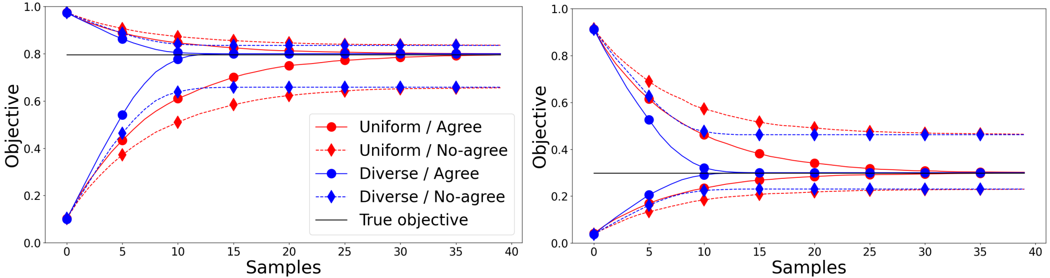
1. Explanations induce a variational surrogate objective
2. A generic algorithm to sample explanations and bounds
3. **Verification of bounds and improved scaling**

Diversity and AGREE step improve bounds



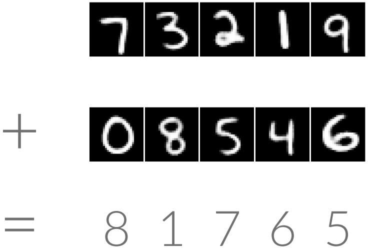


Diversity and AGREE step improve bounds



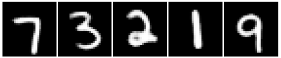
Upshot The AGREE step is crucial to get closer to the true WMC
and diverse samples improves sample efficiency

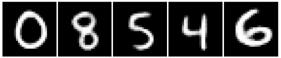
Learning to classify sums of digits



Method	N = 2	N = 4	N = 15
Reference	96.06	92.27	73.97
DeepStochLog	96.40 $\pm$ 0.10	92.70 $\pm$ 0.60	T/O
Embed2Sym	93.81 $\pm$ 1.37	91.65 $\pm$ 0.57	60.46 $\pm$ 20.4
A-NeSI	95.96 $\pm$ 0.38	92.56 $\pm$ 0.79	75.90 $\pm$ 2.21
EXAL	95.82 $\pm$ 0.36	91.77 $\pm$ 0.83	73.27 $\pm$ 2.05
A-NeSI (time)	81.7	198.2	1979.9
EXAL (time)	51.4 $\pm$ 2.3	74.8 $\pm$ 7.2	198.6 $\pm$ 15.7

Learning to classify sums of digits



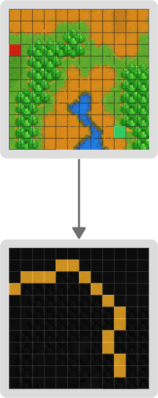
+ 

= 8 1 7 6 5

Method	N = 2	N = 4	N = 15
Reference	96.06	92.27	73.97
DeepStochLog	96.40 ± 0.10	92.70 ± 0.60	T/O
Embed2Sym	93.81 ± 1.37	91.65 ± 0.57	60.46 ± 20.4
A-NeSI	95.96 ± 0.38	92.56 ± 0.79	75.90 ± 2.21
EXAL	95.82 ± 0.36	91.77 ± 0.83	73.27 ± 2.05
A-NeSI (time)	81.7	198.2	1979.9
EXAL (time)	51.4 ± 2.3	74.8 ± 7.2	198.6 ± 15.7

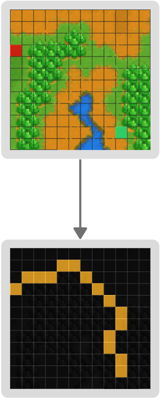
Upshot EXAL learns as well as other methods and is up to **10x faster**

Learning to predict shortest paths



Method	12 × 12	30 × 30
RLOO	43.75 ± 12.35	12.59 ± 16.38
A-NeSI	94.57 ± 2.27	17.13 ± 16.32
A-NeSI + RLOO	98.96 ± 1.33	67.57 ± 36.76
EXAL	94.19 ± 1.74	80.85 ± 3.83
A-NeSI (time)	1380	2640
EXAL (time)	11.1 ± 0.1	84.3 ± 0.7

Learning to predict shortest paths



Method	12 × 12	30 × 30
RLOO	43.75 ± 12.35	12.59 ± 16.38
A-NeSI	94.57 ± 2.27	17.13 ± 16.32
A-NeSI + RLOO	98.96 ± 1.33	67.57 ± 36.76
EXAL	94.19 ± 1.74	80.85 ± 3.83
A-NeSI (time)	1380	2640
EXAL (time)	11.1 ± 0.1	84.3 ± 0.7

Upshot EXAL learns as well or better as other methods and is up to **30x faster**

EXAL learns faster than other methods and provides guarantees

Faster learning	EXAL with diverse explanations outperforms SOTA by always providing a satisfying signal

EXAL learns faster than other methods and provides guarantees

Faster learning	EXAL with diverse explanations outperforms SOTA by always providing a satisfying signal
Provable guarantees	EXAL provides bounds on the true WMC that other methods do not have

EXAL learns faster than other methods and provides guarantees

Faster learning	EXAL with diverse explanations outperforms SOTA by always providing a satisfying signal
Provable guarantees	EXAL provides bounds on the true WMC that other methods do not have
Generic and flexible	EXAL is a generic algorithm and agnostic to the sampler but we provide a universal DPLL-based sampler

Visit personal page

Read our paper