Techniques for Symbol Grounding with SATNet

Sever Topan\textsuperscript{1, 2}, David Rolnick\textsuperscript{1, 3}, Xujie Si\textsuperscript{1, 3}

\textsuperscript{1}McGill University, \textsuperscript{2}NVIDIA, \textsuperscript{3}Mila

October 2021
Introduction: Neurosymbolic Learning

Neural \[\rightarrow\] Symbolic

Merge advances in statistical (neural) models with symbolic knowledge representation and logical reasoning
Introduction: Neurosymbolic Learning

Neural] – [Symbolic

Merge advances in statistical (neural) models with symbolic knowledge representation and logical reasoning

Potential to address limitations in DNN’s:
Introduction: Neurosymbolic Learning

Neural] – [Symbolic

Merge advances in statistical (neural) models with symbolic knowledge representation and logical reasoning

Potential to address limitations in DNN’s:
- Explainability
Introduction: Neurosymbolic Learning

Neural] — [Symbolic

Merge advances in statistical (neural) models with symbolic knowledge representation and logical reasoning

Potential to address limitations in DNN’s:

- Explainability
- Adversarial Robustness
Introduction: Neurosymbolic Learning

Merge advances in statistical (neural) models with symbolic knowledge representation and logical reasoning

Potential to address limitations in DNN’s:
- Explainability
- Adversarial Robustness
- Data Efficiency
Introduction: Neurosymbolic Learning

Neural — Symbolic

Merge advances in statistical (neural) models with symbolic knowledge representation and logical reasoning

Potential to address limitations in DNN’s:

- Explainability
- Adversarial Robustness
- Data Efficiency
- Solve hard logic problems
Introduction: Symbol Grounding

At the interface between a neural and a symbolic module, the meaning of the symbols must be established.
Introduction: Symbol Grounding

At the interface between a neural and a symbolic module, the meaning of the symbols must be established.

This is known as Symbol Grounding.
Prototypical Example: Symbol Grounding in Visual Sudoku

- MNIST digits visually represent input cells

Symbol Grounding: understanding that the shape of the handwritten digit corresponds to one of 9 unique symbols

Two levels of supervision for the problem:
Prototypical Example: Symbol Grounding in Visual Sudoku

- MNIST digits visually represent input cells
- Expected outputs represented numerically
Prototypical Example: Symbol Grounding in Visual Sudoku

- MNIST digits visually represent input cells
- Expected outputs represented numerically
- To solve this problem, the system must:
  - Learn the meanings of digits (neural)
  - Learn the rules of Sudoku (symbolic)
Prototypical Example: Symbol Grounding in Visual Sudoku

- MNIST digits visually represent input cells
- Expected outputs represented numerically
- To solve this problem, the system must:
  - Learn the meanings of digits (neural)
  - Learn the rules of Sudoku (symbolic)

Symbol Grounding: understanding that the shape of the handwritten digit corresponds to one of 9 unique symbols
Prototypical Example: Symbol Grounding in Visual Sudoku

- MNIST digits visually represent input cells
- Expected outputs represented numerically
- To solve this problem, the system must:
  ▶ Learn the meanings of digits (neural)
  ▶ Learn the rules of Sudoku (symbolic)

- **Symbol Grounding**: understanding that the shape of the handwritten digit corresponds to one of 9 unique symbols
- Two levels of supervision for the problem:
Prototypical Example: Symbol Grounding in Visual Sudoku

- MNIST digits visually represent input cells
- Expected outputs represented numerically
- To solve this problem, the system must:
  ▶ Learn the meanings of digits (neural)
  ▶ Learn the rules of Sudoku (symbolic)

**Symbol Grounding**: understanding that the shape of the handwritten digit corresponds to one of 9 unique symbols

- Two levels of supervision for the problem:

\[
\begin{bmatrix}
0 & 8 & 0 \\
0 & 6 & 0 \\
3 & 4 & 0
\end{bmatrix}
\rightarrow
\begin{bmatrix}
1 & 8 & 5 \\
2 & 6 & 9 \\
3 & 4 & 7
\end{bmatrix}
\]

**Grounded Dataset**
Trivial Symbol Grounding

\[
\begin{bmatrix}
1 & 0 & 5 \\
2 & 0 & 9 \\
0 & 0 & 7
\end{bmatrix}
\]

**Ungrounded Dataset**
Difficult Symbol Grounding
Previously, Ungrounded Visual Sudoku was an open problem.
Previously, Ungrounded Visual Sudoku was an open problem

We present a framework for solving Ungrounded Visual MAXSAT problems, like Visual Sudoku, using SATNet (Wang et al. 2019)
Background: SATNet (Wang et al. 2019)

A differentiable MAXSAT solver based on a semidefinite relaxation approach.
Can be integrated into larger DNN pipelines.
Can learn to solve grounded Visual Sudoku, while traditional DNN’s cannot.

Forward Pass: Solves MAXSAT to solve Visual Sudoku.
Background: SATNet (Wang et al. 2019)

- A differentiable MAXSAT solver based on a semidefinite relaxation approach
Background: SATNet (Wang et al. 2019)

- A differentiable MAXSAT solver based on a semidefinite relaxation approach
- Can be integrated into larger DNN pipelines
Background: SATNet (Wang et al. 2019)

- A differentiable MAXSAT solver based on a semidefinite relaxation approach
- Can be integrated into larger DNN pipelines
- Can learn to solve grounded Visual Sudoku, while traditional DNN’s cannot
Background: SATNet Limitations (Chang et al. 2020)

But, SATNet previously could not solve Ungrounded problems, having 0% accuracy
Background: SATNet Limitations (Chang et al. 2020)

- But, SATNet previously could not solve Ungrounded problems, having 0% accuracy
- this issue is known as label leakage
But, SATNet previously could not solve Ungrounded problems, having 0% accuracy.

This issue is known as label leakage.

It limits usefulness of DNN-SATNet hybrid architectures.
Our proposed framework consists of the following steps:

1. Clustering
2. Self-Grounded Training
3. Proofreading
Our proposed framework consists of the following steps:

1. **Clustering**
2. **Self-Grounded Training**
3. **Proofreading**
Method: Clustering

- Intuition: extract semantically relevant aspect of input images using clustering
Method: Clustering

- **Intuition**: extract semantically relevant aspect of input images using clustering
- **Unsupervised pre-training using InfoGAN** (Chen et al. 2016)
Method: Clustering

- Intuition: extract semantically relevant aspect of input images using clustering
- Unsupervised pre-training using InfoGAN (Chen et al. 2016)
- InfoGAN is able to cluster MNIST digits with about 95% accuracy
Aside: Permutation Invariance

- Inputs are clustered with 95% accuracy, but we don’t know which number corresponds to which label.
Aside: Permutation Invariance

- Inputs are clustered with 95% accuracy, but we don’t know which number corresponds to which label
- We cannot use ground truth labels out of the box
Aside: Permutation Invariance

- Inputs are clustered with 95% accuracy, but we don’t know which number corresponds to which label
- We cannot use ground truth labels out of the box
- Example: imagine we had a correct sudoku solver but assigned random labels to our clusters
Aside: Permutation Invariance

- Inputs are clustered with 95% accuracy, but we don’t know which number corresponds to which label
- We cannot use ground truth labels out of the box
- Example: imagine we had a correct sudoku solver but assigned random labels to our clusters

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>8</th>
<th>3</th>
<th>9</th>
<th>4</th>
<th>5</th>
<th>7</th>
<th>6</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>7</td>
<td>9</td>
</tr>
</tbody>
</table>

Table: Two rows of a board predicted by a perfect sudoku model which uses InfoGAN clusters

<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>5</th>
<th></th>
<th>8</th>
<th>9</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8</td>
<td>7</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table: Two rows of the corresponding Ground Truth
Aside: Permutation Invariance

<table>
<thead>
<tr>
<th>1</th>
<th>8</th>
<th>3</th>
<th>9</th>
<th>4</th>
<th>5</th>
<th>7</th>
<th>6</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>5</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>7</td>
<td>9</td>
</tr>
</tbody>
</table>

Table: Two rows of a board predicted by a perfect model which uses InfoGAN.

<table>
<thead>
<tr>
<th>7</th>
<th>5</th>
<th></th>
<th>8</th>
<th>9</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>7</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table: Two rows of the corresponding Ground Truth

- Labels can be different as long as they are consistent
Aside: Permutation Invariance

Table: Two rows of a board predicted by a perfect model which uses InfoGAN.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>8</th>
<th>3</th>
<th>9</th>
<th>4</th>
<th>5</th>
<th>7</th>
<th>6</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>5</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>7</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

Table: Two rows of the corresponding Ground Truth

<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>5</th>
<th></th>
<th>8</th>
<th>9</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Labels can be different as long as they are consistent
Aside: Permutation Invariance

Table: Two rows of a board predicted by a perfect model which uses InfoGAN.

<table>
<thead>
<tr>
<th>1</th>
<th>8</th>
<th>3</th>
<th>9</th>
<th>4</th>
<th>5</th>
<th>7</th>
<th>6</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>5</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>7</td>
<td>9</td>
</tr>
</tbody>
</table>

Table: Two rows of the corresponding Ground Truth

<table>
<thead>
<tr>
<th>7</th>
<th>5</th>
<th></th>
<th>8</th>
<th>9</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>7</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Labels can be different as long as they are **consistent**
- This applies to other SAT-solvable games, beyond Sudoku
Aside: Permutation Invariance

Table: Two rows of a board predicted by a perfect model which uses InfoGAN.

Table: Two rows of the corresponding Ground Truth

- Labels can be different as long as they are consistent
- This applies to other SAT-solvable games, beyond Sudoku
- Common loss functions, such as $l_2$ norm or binary cross-entropy (BCE), will not work
Aside: Permutation Invariance

Table: Two rows of a board predicted by a perfect model which uses InfoGAN.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>8</th>
<th>3</th>
<th>9</th>
<th>4</th>
<th>5</th>
<th>7</th>
<th>6</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>5</td>
<td>8</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>7</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

Table: Two rows of the corresponding Ground Truth

<table>
<thead>
<tr>
<th></th>
<th>7</th>
<th>5</th>
<th></th>
<th>8</th>
<th>9</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Labels can be different as long as they are **consistent**
- This applies to other SAT-solvable games, beyond Sudoku
- Common loss functions, such as \( l_2 \) norm or binary cross-entropy (BCE), will not work
- Need a different loss function
Our proposed framework consists of the following steps:

1. Clustering
2. **Self-Grounded Training**
3. Proofreading
Method: Self-Grounded Training

Introduce the Symbol Grounding Loss (SGL):

\[
\mathcal{L}(\hat{y}_{out}^{PTE}, y^{LE}) := 1 - \max_j \left( \exp[-(y^{LE}(j), \hat{y}_{out}^{PTE}(i))] \right) \]

Learn P, and Train End-to-End
Method: Self-Grounded Training

Introduce the Symbol Grounding Loss (SGL):

\[
\mathcal{L}(\hat{y}_{out}^{PTE}, y^{LE}) := 1 - i \left( \max_{j} \exp[-(y^{LE}(j), \hat{y}_{out}^{PTE}(i))] \right),
\]

- SGL infers a permutation matrix between predictions and labels (See paper for further details)
Method: Self-Grounded Training

Introduce the Symbol Grounding Loss (SGL):

\[ \mathcal{L}(\hat{y}_\text{out}^{PTE}, y^{LE}) := 1 - i \max_j (\exp[-(y^{LE}(j), \hat{y}_\text{out}^{PTE}(i))])} \),

- SGL infers a permutation matrix between predictions and labels (See paper for further details)
- System is trained end-to-end under the Symbol Grounding Loss
Method: Self-Grounded Training

Introduce the Symbol Grounding Loss (SGL):

$$\mathcal{L}(\hat{y}_{out}^{PTE}, y^{LE}) := 1 - \max_j (\exp[-(y^{LE}(j), \hat{y}_{out}^{PTE}(i))])$$,

- SGL infers a permutation matrix between predictions and labels (See paper for further details)
- System is trained end-to-end under the Symbol Grounding Loss
- A permutation matrix $P$ is implicitly learned by SGL
Introduce the Symbol Grounding Loss (SGL):

$$\mathcal{L}(\hat{y}_{out}^{PTE}, y^{LE}) := 1 - i \left( \max_j \exp[-(y^{LE}(j), \hat{y}_{out}^{PTE}(i))] \right).$$

- SGL infers a permutation matrix between predictions and labels (See paper for further details)
- System is trained end-to-end under the Symbol Grounding Loss
- A permutation matrix $P$ is implicitly learned by SGL
- Once $P$ has converged, continue training under standard BCE
Our proposed framework consists of the following steps:

1. **Clustering**
2. **Self-Grounded Training**
3. **Proofreading**
Method: Proofreading

- Insert a linear layer before SATNet
Method: Proofreading

- Insert a linear layer before SATNet
- Initialize to a slightly noisy identity transform
Method: Proofreading

- Insert a linear layer before SATNet
- Initialize to a slightly noisy identity transform
- Freeze rest of system, train proofreader
Method: Proofreading

- Insert a linear layer before SATNet
- Initialize to a slightly noisy identity transform
- Freeze rest of system, train proofreader
- Improves accuracy marginally in both our method and prior SATNet architectures
Our proposed framework consists of the following steps:

1. Clustering
2. Self-Grounded Training
3. Proofreading
### Results: Ungrounded Visual Sudoku

<table>
<thead>
<tr>
<th>Model Configuration</th>
<th>Grounded vs. Ungrounded Data</th>
<th>Total Board Accuracy (%)</th>
<th>Per-Cell Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original SATNet</td>
<td>grounded</td>
<td>66.5 ± 1.0</td>
<td>98.8 ± 0.1</td>
</tr>
<tr>
<td>Original SATNet</td>
<td>ungrounded</td>
<td>0 ± 0.0</td>
<td>11.2 ± 0.1</td>
</tr>
<tr>
<td>Our Method</td>
<td>ungrounded</td>
<td>64.8 ± 3.0</td>
<td>98.4 ± 0.2</td>
</tr>
</tbody>
</table>
### Results: Effect of Proofreader

<table>
<thead>
<tr>
<th>Model Configuration</th>
<th>Proofreader Present?</th>
<th>Total Board Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Non-visual</td>
<td>no</td>
<td>96.6 ± 0.3</td>
</tr>
<tr>
<td>Original Non-visual</td>
<td>yes</td>
<td>97.1 ± 0.3</td>
</tr>
<tr>
<td>Original Visual</td>
<td>no</td>
<td>66.5 ± 1.0</td>
</tr>
<tr>
<td>Original Visual</td>
<td>yes</td>
<td>67.6 ± 1.2</td>
</tr>
<tr>
<td>Our Method</td>
<td>no</td>
<td>62.8 ± 3.2</td>
</tr>
<tr>
<td>Our Method</td>
<td>yes</td>
<td>64.8 ± 3.0</td>
</tr>
</tbody>
</table>
Limitations & Future Work

- In an ablation test, we find that our system requires roughly at least 88% clustering accuracy in order for the rest of the pipeline to progress.
Limitations & Future Work

- In an ablation test, we find that our system requires roughly at least 88% clustering accuracy in order for the rest of the pipeline to progress.
- Our approach requires prior knowledge of the number of symbols.

Above can be alleviated but Symbol Grounding Loss supporting a general surjective mapping instead of permutation.
Limitations & Future Work

- In an ablation test, we find that our system requires roughly at least 88% clustering accuracy in order for the rest of the pipeline to progress.
- Our approach requires prior knowledge of the *number* of symbols.
- Above can be alleviated but Symbol Grounding Loss supporting a general surjective mapping instead of permutation.
Conclusion

In this work we:

- Distinguish between grounded and ungrounded variants of Visual MAXSAT problems
- Present a framework which enables SATNet to solve ungrounded datasets
- New state-of-the-art for Ungrounded Visual Sudoku, previously 0%
- Describe a proofreading methodology which improves both our architecture and prior models

Available: github.com/SeverTopan/SATNet
Conclusion

In this work we:

- Distinguish between grounded and ungrounded variants of Visual MAXSAT problems
Conclusion

In this work we:

- Distinguish between grounded and ungrounded variants of Visual MAXSAT problems
- Present a framework which enables SATNet to solve ungrounded datasets

Available: github.com/SeverTopan/SATNet
Conclusion

In this work we:

- Distinguish between grounded and ungrounded variants of Visual MAXSAT problems
- Present a framework which enables SATNet to solve ungrounded datasets
- New state-of-the-art for Ungrounded Visual Sudoku, previously 0%
Conclusion

In this work we:

- Distinguish between grounded and ungrounded variants of Visual MAXSAT problems
- Present a framework which enables SATNet to solve ungrounded datasets
- New state-of-the-art for Ungrounded Visual Sudoku, previously 0%
- Describe a proofreading methodology which improves both our architecture and prior models
Conclusion

In this work we:

- Distinguish between grounded and ungrounded variants of Visual MAXSAT problems
- Present a framework which enables SATNet to solve ungrounded datasets
- New state-of-the-art for Ungrounded Visual Sudoku, previously 0%
- Describe a proofreading methodology which improves both our architecture and prior models
- Available: github.com/SeverTopan/SATNet
Techniques for Symbol Grounding with SATNet

Sever Topan$^1, 2$, David Rolnick$^1, 3$, Xujie Si$^1, 3$

$^1$McGill University, $^2$NVIDIA, $^3$Mila

October 2021