Compositional Visual Reasoning: Interpretability and Evaluation
Sanjay Subramanian, with many collaborators
Recent developments in compositional visual reasoning

VQA datasets

Natural Language

Natural Images

Compositionality

VQA (Agrawal et al., 2017)
Recent developments in compositional visual reasoning

VQA datasets

- Natural Language
- Natural Images
- Compositionality

CLEVR (Johnson et al., 2017)

VQA (Agrawal et al., 2017)
Recent developments in compositional visual reasoning

VQA datasets

- NLVR (Suhr et al., 2017)
- CLEVR (Johnson et al., 2017)
- VQA (Agrawal et al., 2017)
- GQA (Hudson and Manning 2019)
Recent developments in compositional visual reasoning

VQA datasets

- NLVR2 (Suhr et al., 2019)
- NLVR (Suhr et al., 2017)
- CLEVR (Johnson et al., 2017)
- VQA (Agrawal et al., 2017)
- GQA (Hudson and Manning 2019)
Recent developments in compositional visual reasoning

- Early VQA datasets were either simple and natural (e.g. VQA; Agrawal et al. 2017) or compositional and synthetic (e.g. CLEVR; Johnson et al. 2017)
- Recent compositional datasets:
  - GQA (Hudson and Manning 2019) -- synthetic question with natural image. Classification and open-ended questions.
Recent developments in compositional visual reasoning

- Large-scale pre-trained transformers have been successful
- Example: LXMERT (Tan and Bansal, 2019)

\[
\text{LXMERT loss} = \text{Masked language-modeling} + \text{Visual feature regression} + \text{Image-text matching} + \text{VQA}
\]

- Requires paired images and captions (COCO/Visual Genome) and VQA data
- SOTA on NLVR2, strong performance on GQA
Performance Gains from Pre-training

NLVR2 (Suhr et al. 2019) Test Accuracy

- Baseline
- VisualBERT (Li et al. 2019)
- LXMERT (Tan and Bansal 2019)
- UNITER (Chen et al. 2019)
Issues raised by large pre-trained models

1. Interpretability: Can we make these models interpretable?
   ○ Unclear how to extract the steps of a vanilla Transformer
   ○ Particularly salient for compositional tasks

2. Evaluation: Are there shortcuts in these compositional datasets that enable models to perform well without going through the apparent reasoning steps?
   ○ Specifically: is object+attribute detection sufficient?
Obtaining Faithful Interpretations from Compositional Neural Networks

SS*, Ben Bogin*, Nitish Gupta*

Tomer Wolfson, Sameer Singh, Jonathan Berant, Matt Gardner
ACL 2020
Compositional reasoning

All dogs are black
Compositional reasoning

All dogs are black

LXMERT (black-box neural network)

False

Tan and Bansal, EMNLP 2019
Compositional reasoning

All dogs are black

LXMERT (black-box neural network)

False

Tan and Bansal, EMNLP 2019

Not Interpretable
Neural Module Networks (NMN)

All dogs are black

[Andreas et al., NAACL 2016]
Neural Module Networks (NMN)

All dogs are black

100%

100%

parse

execute

equal

count

find[dogs]

filter[black]
All dogs are black

Neural Module Networks (NMN)
Neural Module Networks (NMN)

All dogs are black

100% 100%

execute

equal

2 1

parse

count count

filter[black]

find[dogs]

find[dogs]
Neural Module Networks (NMN)

All dogs are black

execute

False

equal

2

count

1

count

filter[black]

find[dogs]

find[dogs]

parse

100%

100%
Neural Module Networks (NMN)

All dogs are black

Find dogs

Filter [black]

Count equal 1

Parse True execute

Count 2

Modules
Learnable NNs to perform atomic tasks
Neural Module Networks (NMN)

All dogs are black

Modules
Learnable NNs to perform atomic tasks

Count
2

Equal
False

Filter[black]

Find[dogs]

Count
1

Backpropagation

Learn parameters for all modules based on the answer as weak signal

Parse

Execute
Neural Module Networks (NMN)

All dogs are black

Interpretable!
Module execution is not faithful!

All dogs are black

Module execution is not faithful to its intended reasoning

After training using only end-task supervision

execute

equal

count

parse

find[dogs]

filter[black]

count
Faithful module execution

- Module performs its intended operation; hence faithful
- Module does not perform its intended operation; hence not-faithful
Module execution is not faithful to its intended reasoning.

All dogs are black

After training using only end-task supervision

Program is not a faithful explanation of the model behavior
In this work ...

We propose,

(1) Ways to improve module-wise faithfulness

(2) Systematic evaluation of intermediate module execution
In this work ...

We propose,

(1) Ways to improve module-wise faithfulness

(2) Systematic evaluation of intermediate module execution
What’s causing the unfaithful interpretations?

- Model gets high accuracy but low faithfulness → multiple reasoning steps are being collapsed within one module (or in the contextualizing model)

- Possible causes for the collapsed reasoning:
  - Count architecture is too expressive
  - Contextualized representations already reflect the reasoning

- Supervising module outputs directly is another method
Dataset and Implementation

**NLVR2 (Suhr et al., 2019)**

two dogs are touching a food dish with their face

Train and evaluate on examples with QDMR program annotation
~32,000 examples

[BREAK; Wolfson et al. 2020]

**Module List:**

- Find() → ObjectSet
- Filter(ObjectSet) → ObjectSet
- Relation(ObjectSet, ObjectSet) → ObjectSet
- Project(ObjectSet) → ObjectSet
- Count(ObjectSet) → number
- Parameter-less: Equals, Greater-than, etc.
- Macros: In-each-image, In-at-least-one-image
1) Visual-NMN: Count module mediates backprop

all dogs are black

```
find[dogs] -> count
filter[black] -> count
```

execute

```
equal
False
```
1) Visual-NMN: Lower-capacity Count Module improves faithfulness

Layer-count
1.93
(many parameters)

Count

there are three seals in the image pair.
→ Answer: False
1) Visual-NMN: Lower-capacity Count Module improves faithfulness

Layer-count

1.93

(Many parameters)

Count

76%

find[seals]

50%

find[seals]

there are three seals in the image pair. → Answer: False

Sum-count

1.10

(no parameters)

Count (+)

70%

find[seals]

40%

find[seals]

0%

find[seals]
1) Visual-NMN: Lower-capacity Count Module improves faithfulness

Layer-count
- Count
- 1.93
- (many parameters)
- 76% find[seals] 50% find[seals]

Graph-count (Zhang et al., 2018)
- Count
- 1.97
- (few parameters)
- 97% find[seals] 97% find[seals] 2% find[seals]

Sum-count
- Count (+)
- 1.10
- (no parameters)
- 70% find[seals] 40% find[seals] 0% find[seals]

There are three seals in the image pair. → Answer: False
2) Decontextualized Word Vectors Improve Faithfulness

*the llamas in both images are eating*

*doesn't find llamas*
2) Decontextualized Word Vectors Improve Faithfulness

The llamas in both images are eating doesn't find llamas

Find [llamas]

The llamas in both images are eating

LXMERT

The llamas in both images are eating

doesn't find llamas
2) Decontextualized Word Vectors Improve Faithfulness

LXMERT

the llamas in both images are eating

doesn't find llamas, effectively searching for eating llamas
2) Decontextualized Word Vectors Improve Faithfulness

doesn't find llamas, effectively searching for eating llamas

Correctly finds the llamas

LXMERT

the llamas in both images are eating

the llamas in both images are eating

LXMERT
3) Supervising module output improves faithfulness

Pre-train find and filter with auxiliary module supervision on different dataset (GQA)

Is there a green apple?

Auxiliary supervision:

```
+1   -1
exist
+1   -1
filter[green]
find[apple]
```

[Hudson and Manning, 2019]
In this work ...

We propose,

(1) Ways to improve module-wise faithfulness

(2) Systematic evaluation of intermediate module execution
Previous work

We propose,

1. Ways to improve module-wise faithfulness
2. Systematic evaluation of intermediate module execution

[Andreas et al. 2016] [Hu et al. 2017]
One exception in previous work: Hu et al. 2018 asks humans to evaluate module outputs in two ways:

- Subjective understanding: Rate (on a 4-point scale) how well you can understand the model’s reasoning via the module outputs
- Forward prediction: Predict the model’s output and failure based on the module outputs

Our approach allows evaluation of multiple models without any additional annotations.
How do we evaluate faithfulness?

two dogs are touching a food dish with their face

Gold Program

equal

count

with-relation [is touching]

relocate [face]

find [dog]

find [food dish]

number [two]
How do we evaluate faithfulness?

**Gold Program**

```
equal
count
with-relation [is touching]
relocate [face]
  find [dog]
find [food dish]
number [two]
```

We collect intermediate outputs for 536 programs.

two dogs are touching a food dish with their face
How do we evaluate faithfulness?

two dogs are touching a food dish with their face

Gold Program

- equal
- count
- with-relation [is touching]
  - relocate [face]
    - find [dog]
  - find [food dish]
- number [two]

We collect intermediate outputs for 536 programs

Compute precision, recall, $F_1$
How do we evaluate faithfulness?

**Gold Program**
- equal
- count
- with-relation [is touching]
  - relocate [face]
  - find [dog]
  - find [food dish]
- number [two]

**prediction**

**gold**

two dogs are touching a food dish with their face

$F_1: 0.5$

We collect intermediate outputs for 536 programs

Compute precision, recall, $F_1$
Results - NLVR2

Accuracy

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>LXMERT</td>
<td>71.7</td>
</tr>
<tr>
<td>NMN</td>
<td>71.2</td>
</tr>
</tbody>
</table>

Faithfulness ($F_1$)

(average across modules)
Results - NLVR2

Accuracy:
- LXMERT: 71.7
- NMN: 71.2
- +GraphCount: 69.6

Faithfulness ($F_1$):
- NMN: 0.11
- +GraphCount: 0.28

(average across modules)
Results - NLVR2

**Accuracy**

- LXMERT: 71.7
- NMN: 71.2
- +GraphCount: 69.6
- +Decont.: 67.3

**Faithfulness (F₁)**

- NMN: 0.11
- +GraphCount: 0.28
- +Decont.: 0.39

(average across modules)
Results - NLVR2

Accuracy

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>LXMERT</td>
<td>71.7</td>
</tr>
<tr>
<td>NMN</td>
<td>71.2</td>
</tr>
<tr>
<td>+GraphCount</td>
<td>69.6</td>
</tr>
<tr>
<td>+Decont.</td>
<td>67.3</td>
</tr>
<tr>
<td>+Pre-training</td>
<td>68.7</td>
</tr>
</tbody>
</table>

Faithfulness ($F_1$)

<table>
<thead>
<tr>
<th>Model</th>
<th>Faithfulness</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMN</td>
<td>0.11</td>
</tr>
<tr>
<td>+GraphCount</td>
<td>0.28</td>
</tr>
<tr>
<td>+Decont.</td>
<td>0.39</td>
</tr>
<tr>
<td>+Pre-training</td>
<td>0.47</td>
</tr>
</tbody>
</table>

(average across modules)
Results - DROP

Accuracy ($F_1$)

Faithfulness (cross-entropy; lower is better)

(average across modules)
Results - DROP

Accuracy ($F_1$)

<table>
<thead>
<tr>
<th></th>
<th>Text-NMN</th>
<th>+ModuleSupervision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy ($F_1$)</td>
<td>65.3</td>
<td>65.7</td>
</tr>
</tbody>
</table>

Faithfulness (cross-entropy; lower is better)

<table>
<thead>
<tr>
<th></th>
<th>Text-NMN</th>
<th>+ModuleSupervision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faithfulness</td>
<td>11.2</td>
<td>6.5</td>
</tr>
</tbody>
</table>

(average across modules)
"a small white puppy is laying down with its tongue out."
"a small white puppy is laying down with its tongue out."
"a small white puppy is laying down with its tongue out."
"a small white puppy is laying down with its tongue out."

Gold Program

filter [laying down]  
filter [white]  
filter [small]  
find [puppy]
"a small white puppy is laying down with its tongue out."

**Gold Program**

- project [tongue]
  - filter [laying down]
  - filter [white]
    - filter [small]
    - find [puppy]
"a small white puppy is laying down with its tongue out."

**Gold Program**

```plaintext
filter [is out]
  project [tongue]
  filter [laying down]
  filter [white]
  filter [small]
  find [puppy]
```
"a small white puppy is laying down with its tongue out."

Gold Program

Exist True

filter [is out]
project [tongue]
filter [laying down]
filter [white]
filter [small]
find [puppy]
Another example of interpretable compositional reasoning: Grounded Chart Parser

What is the size of the small matte object that is behind the large shiny block and is the same color as the tiny shiny sphere?

Ben Bogin,
SS, Matt Gardner,
Jonathan Berant,
Accepted to TACL
## Grounded Chart Parser Results

<table>
<thead>
<tr>
<th></th>
<th>CLEVR</th>
<th>CLOSURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC</td>
<td>98.5</td>
<td>72.4</td>
</tr>
<tr>
<td>FiLM</td>
<td>97.0</td>
<td>60.1</td>
</tr>
<tr>
<td><strong>GLT (our model)</strong></td>
<td><strong>99.1</strong></td>
<td><strong>96.1 ( \pm 2.5 )</strong></td>
</tr>
<tr>
<td>NS-VQA † ‡</td>
<td>100</td>
<td>77.2</td>
</tr>
<tr>
<td>PG+EE (18K prog.) †</td>
<td>95.4</td>
<td>-</td>
</tr>
<tr>
<td>PG-Vector-NMN †</td>
<td>98.0</td>
<td>71.3</td>
</tr>
<tr>
<td>GT-Vector-NMN † ‡</td>
<td>98.0</td>
<td>94.4</td>
</tr>
</tbody>
</table>

### Accuracy

<table>
<thead>
<tr>
<th>Constituents Recall (%)</th>
<th>CLEVR</th>
<th>CLOSURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denotation (F1)</td>
<td>95.9%</td>
<td>94.7</td>
</tr>
</tbody>
</table>

### Interpretability

<table>
<thead>
<tr>
<th>Constituents Recall (%)</th>
<th>CLEVR</th>
<th>CLOSURE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denotation (F1)</td>
<td>95.9%</td>
<td>94.7</td>
</tr>
</tbody>
</table>
Evaluation: Are pre-trained systems doing compositional reasoning?

NLVR2 Example: “The dog in the image on the right is wearing a collar.”
Are pre-trained systems doing compositional reasoning?

NLVR2 Example: “The dog in the image on the right is wearing a collar.”

Label: False

Label: True

Harder image (Not Taken from NLVR2)
Are pre-trained systems doing compositional reasoning?

NLVR2 Example: “The dog in the image on the right is wearing a collar.”

Relation “wearing” is not necessary to answer these correctly

Label: False

Label: True

Harder image (Not Taken from NLVR2)
Experiment: Remove relational cues

- Mask/drop prepositions and verbs across all sentences
- LXMERT’s Performance is nearly the same!
- Similar result on GQA

Example

[CLS] the dog **in** the image **on** the right **is wearing** a collar. [SEP]
Experiment: Input Reduction

Remove token from NLVR2 sentence with least gradient iteratively without changing prediction on any image pair (Feng et al. 2018)

Examples

[CLS] a silver spoon has cookie dough in it...[SEP]

[CLS] at least one human is wearing eye glasses...[SEP]

[CLS] the left and right image contains no more than three bottles of lot ##ion...[SEP]

Token Sequence lengths before and after Input Reduction for Correctly Answered Instances
Experiment: Syntax Probe

- Compositionality presumably requires some knowledge of syntax
- How well does LXMERT encode syntax trees?
- Structural probe (Hewitt and Manning 2019) learns to map from encoder representations to pairwise parse-tree distance

![Graph showing correlation between parse-tree distance and predicted distance]

- BERT Last Layer
- LXMERT Last Language Layer
- LXMERT Last Cross-modal layer
Evaluation: Contrast sets for NLVR2

- What happens when we modify slightly the input language or images for NLVR2?
- Contrast sets: non-i.i.d. test data for many NLP tasks to evaluate how well models do around local decision boundaries

Matt Gardner and many others, EMNLP-Findings 2020
Evaluation: Contrast sets for NLVR2

- What happens when we modify slightly the input language or images for NLVR2?
- Contrast sets: non-i.i.d. test data for many NLP tasks to evaluate how well models do around local decision boundaries
- NLVR2 Results (for LXMERT):

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td># of Examples</td>
<td>994</td>
</tr>
<tr>
<td># of Sets</td>
<td>479</td>
</tr>
<tr>
<td>Original Test Accuracy</td>
<td>76.4</td>
</tr>
<tr>
<td>Contrast Test Accuracy</td>
<td>61.1 (-15.3)</td>
</tr>
<tr>
<td>Consistency</td>
<td>30.1</td>
</tr>
</tbody>
</table>

Matt Gardner and many others, EMNLP-Findings 2020
Evaluation: Contrast sets for NLVR2

Example:

Two **similarly-colored** and similarly-posed chow dogs are face to face in one image.

Two **differently-colored** but similarly-posed chow dogs are face to face in one image.
Example:

Two similarly-colored and similarly-posed chow dogs are face to face in one image.
Conclusion

- **Interpretability**: Interpretability is still feasible using previous methods (e.g. NMNs) on top of recent pre-trained models
  - Our work relies heavily on gold programs; how well can we do without them?
- **Evaluation**: Pre-training seems to be very good for grounding nouns and adjectives (and perhaps for counting), but relations seem to need more work
Vision+Language in Scientific Documents

MedICaT: A Dataset of Medical Images, Captions, and Textual References (EMNLP-Findings 2020)

Unique features:

- Subfigure-subcaption alignment annotations for > 2000 figures
- Figure references in main body text for > 70% of figures

**FIGURE 1.** The tumor (approximately 40mm in diameter) was hypovascular on enhanced computed tomography scan (right), indicated low intensity on T1-weighted MRI (center), and high intensity on T2-weighted or diffusion MRI (left). Dynamic study revealed peripheral enhancement on a late phase. The tumor located close to the inferior vena cava. MRI = magnetic resonance imaging.

**Corresponding inline reference:** The tumor was hypovascular on enhanced CT scan (**Figure 1**) and indicated low intensity on T1-weighted magnetic resonance imaging (MRI) and high intensity on T2-weighted or diffusion MRI (**Figure 1**).
Collaborators

+many others
1. We propose the concept of module-wise faithfulness and ways to systematically evaluate faithfulness in Visual and Text NMN.
1. We propose the concept of module-wise faithfulness and ways to systematically evaluate faithfulness in Visual and Text NMN.

2. We propose various ways to improve module-wise faithfulness in NMNs.
Conclusion

1. We propose the concept of module-wise faithfulness and ways to systematically evaluate faithfulness in Visual and Text NMN.

2. We propose various ways to improve module-wise faithfulness in NMNs.

3. We release over 700 human-annotated programs with intermediate outputs for NLVR2 and DROP to measure module-wise faithfulness.
1. We propose the concept of module-wise faithfulness and ways to systematically evaluate faithfulness in Visual and Text NMN.

2. We propose various ways to improve module-wise faithfulness in NMNs.

3. We release over 700 human-annotated programs with intermediate outputs for NLVR2 and DROP to measure module-wise faithfulness.

Code and annotations: [github.com/allenai/faithful-nmn](https://github.com/allenai/faithful-nmn)
Neural Module Networks for Text Reasoning

Neural networks with learnable parameters to solve an atomic task

Modules for Visual Reasoning

**find[arg]** Find bounding boxes corresponding to "arg"

**filter[condition]** Filter input bounding boxes based on the "condition"

**count** Count the input number of boxes
Two dogs example with what we’re evaluating

two dogs are touching a food dish with their face

<table>
<thead>
<tr>
<th></th>
<th>Program</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>equal</td>
<td>True</td>
</tr>
<tr>
<td></td>
<td>count</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>with-relation [is touching] [2, 5]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>relocate [face] [2, 5]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>find [dog] [1, 4]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>find [food dish] [3, 6]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>number [two]</td>
<td>2</td>
</tr>
</tbody>
</table>

Person 2
Improvement 1: Architectural choice

- Visual-NMN: Count module occurs in every program
  - Layer-count (most flexible): count = FFNN(box probabilities, box representations)

```
find[people]  utt: “there are three people”
```

35%  34%  60%  60%
Improvement 1: Architectural choice

- Visual-NMN: Count module occurs in every program
  - Layer-count (most flexible): count = FFNN(box probabilities, box representations)
    - find [people]  
    - utterance: “there are three people”
  - Sum-count (least flexible): count = Sum(box probabilities)
Improvement 1: Architectural choice

- Visual-NMN: Count module occurs in every program
  - Layer-count (most flexible): count = FFNN(box probabilities, box representations)
  - Sum-count (least flexible): count = Sum(box probabilities)
  - Graph-count: Like Sum-count but accounts for box overlap (Zhang et al., 2018)

\[ \text{find}[\text{people}] \quad \text{utt: "there are three people"} \]
Improvement 1: Architectural choice

- **Visual-NMN**: Count module occurs in every program
  - Layer-count (most flexible): \( \text{count} = \text{FFNN} (\text{box probabilities}, \text{box representations}) \)
  - Sum-count (least flexible): \( \text{count} = \text{Sum} (\text{box probabilities}) \)
  - Graph-count: Like Sum-count but accounts for box overlap (Zhang et al., 2018)

- **Text-NMN**: “extract-answer” module produces a direct answer without compositional reasoning
  - Can improve accuracy by handling reasoning out of scope of modules
  - Decreases faithfulness by collapsing several reasoning steps
Improvement 2: Supervising module output

- Include loss term for individual module outputs
- Visual-NMN: Supervise object box probabilities
  - Module-wise annotations are not available for NLVR2
  - We pre-train on GQA (Hudson et al., 2019) for which we can obtain annotations
- Text-NMN: Supervise token probabilities
  - We use heuristics (proposed by Gupta et al., 2020) to obtain gold spans for `find-num` and `find-date`
Improvement 3: Decontextualized Word Vectors

- Visual NMN: each module uses an attention over tokens to obtain a weighted average of LXMERT (Tan and Bansal, 2019) token representations.
- However, LXMERT’s outputs are already contextualized, so tokens outside the attention can still contribute to the attended representation.
Improvement 3: Decontextualized Word Vectors

- Visual NMN: each module uses an attention over tokens to obtain a weighted average of LXMERT (Tan and Bansal, 2019) token representations.
- However, LXMERT’s outputs are already contextualized, so tokens outside the attention can still contribute to the attended representation.
- Our proposal: Run LXMERT separately for each module, masking out all tokens outside the module’s utterance attention.

Example: All the dogs are black.

```
dogs  LXMERT  find
black  filter
```
Visual-NMN: Lower-capacity Count Module improves faithfulness

Layer-count

1.95

(find[dog] 50%)

(find[dog] 55%)

(find[dog])

(find[dog])
Visual-NMN: Lower-capacity Count Module improves faithfulness

Layer-count
1.95
(many parameters)

Count

- find[dog] 50%
- find[dog] 55%

Capacity

Sum-count
1.88
(no parameters)

Count (+)

- find[dog] 97%
- find[dog] 91%

Faithfulness
**Visual-NMN: Lower-capacity Count Module improves faithfulness**

- **Layer-count**
  - Count Module: 1.95
  - (many parameters)

- **Graph-count (Zhang et al., 2018)**
  - Count Module: (few parameters)

- **Sum-count**
  - Count Module: 1.88
  - (no parameters)

The diagram illustrates the relationship between capacity and faithfulness, with visual representations of the modules and associated percentages for finding dogs.