Iterated learning for emergent systematicity in VQA


Ankit Vani


Max Schwarzer


Yuchen Lu


Eeshan Dhekane


Aaron Courville

Mila

Université de Montréal

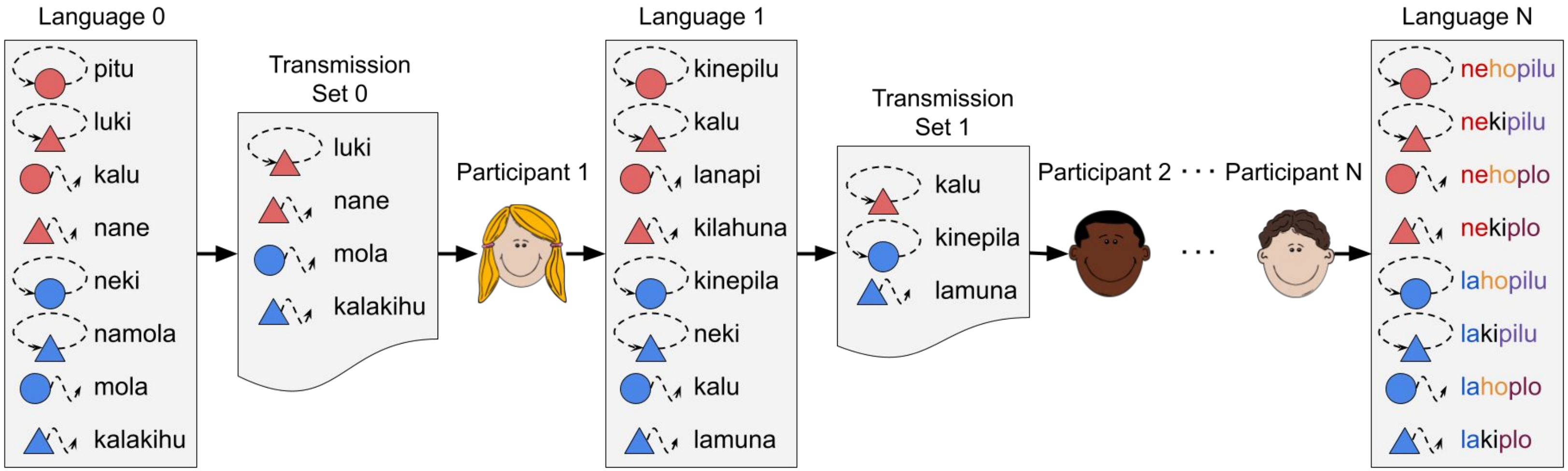
Background

Iterated learning (IL)

- [Kirby et al., 2014] The process by which a behavior arises in one individual
  - Through induction, based on observations of behavior in another individual
    - who acquired that behavior in the same way

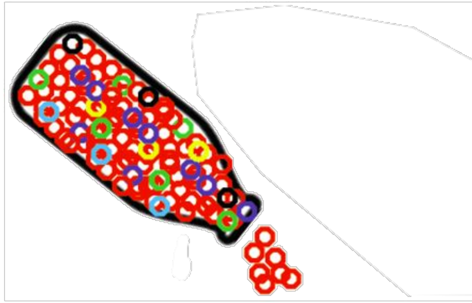
- Leads to the emergence of compositionality in human languages

Iterated learning (IL)



(based on Kirby et al., 2008; 2014)

Learning bottleneck



- Need to learn an **highly expressive** language
  - Through **limited supervision**

- Language properties likely to pass through this bottleneck become universal
  - Compositional rules are more likely to survive the transmission (Kirby, 2001)
    - Easier to learn
    - Faster to learn

  - Language properties depend on prior / inductive bias of agents

Iterated learning in machine learning

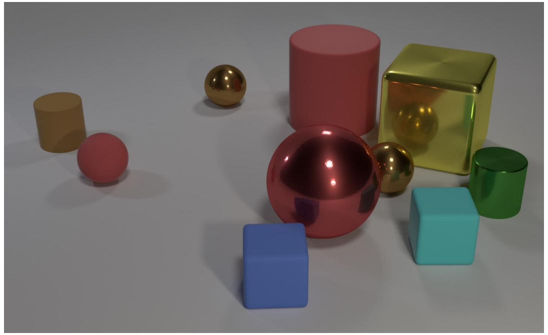
- IL in machine learning has stayed close to its cognitive science roots
    - Most of the work involves agents playing very simple referential games

      (Guo et al., 2019; Li et al., 2019; Cogswell et al., 2019; Dagan et al., 2020; Ren et al., 2020)

- **Claim**: Learning bottleneck is a fundamental way to recover structure

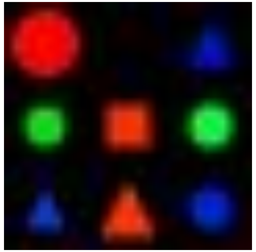
- We demonstrate this in a more complex task of **visual question-answering**

Visual question-answering (VQA)



(Image taken from CLEVR (Johnson et al., 2017))

Q: Are there an equal number of large things and metal spheres?



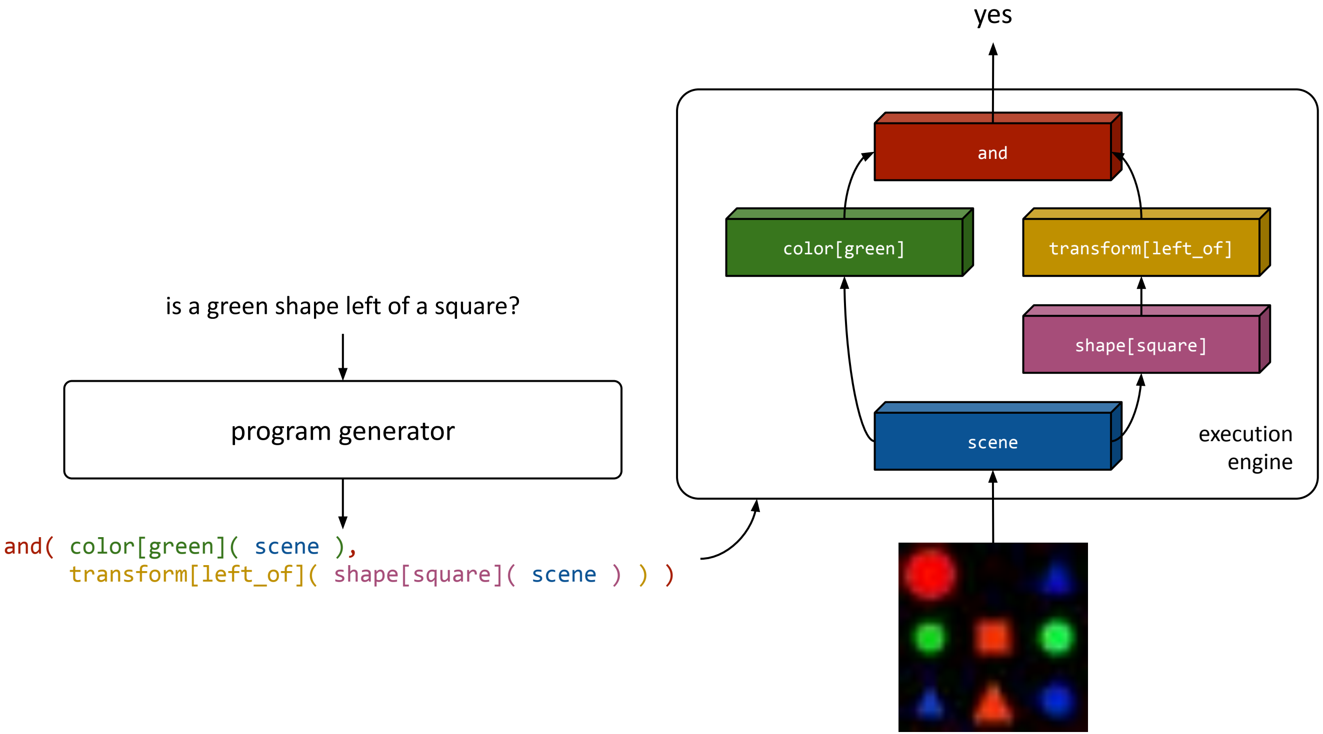
(Image taken from SHAPES (Andreas et al., 2016))

Q: Is a green shape left of a square?

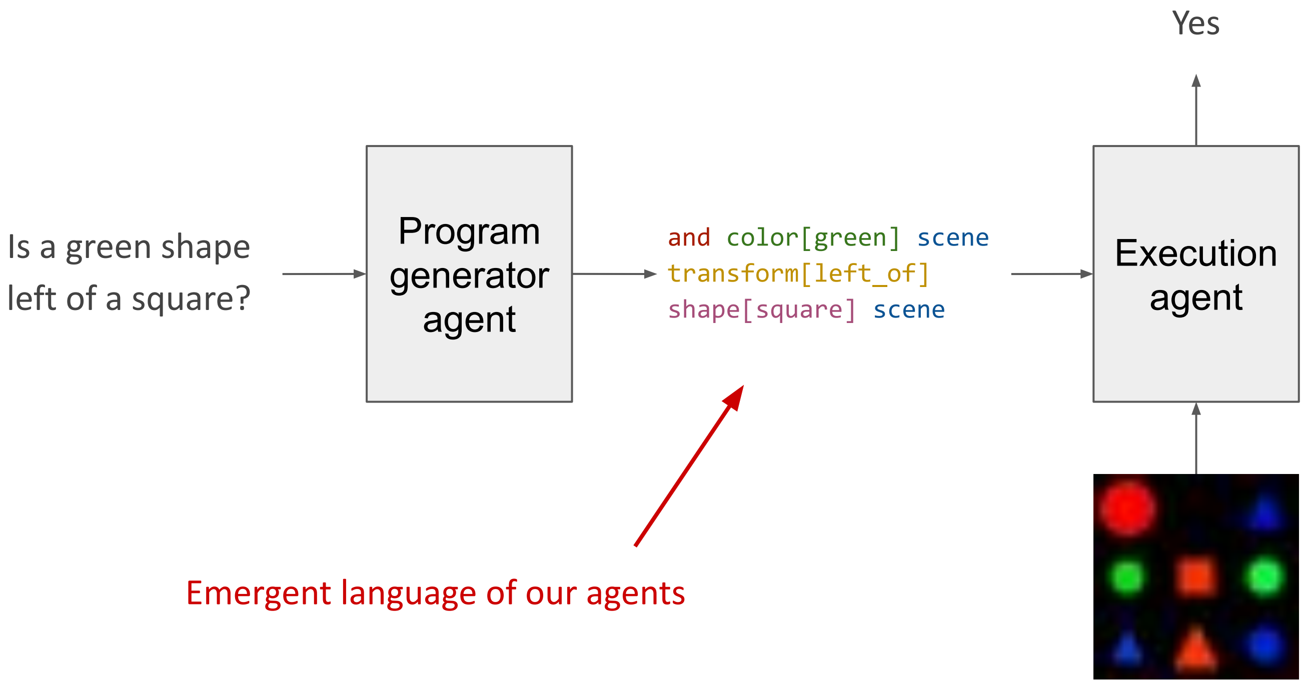
Program:

```
and( color[green]( scene ),
     transform[left_of]( shape[square]( scene ) ) )
```

Neural module networks (NMNs)



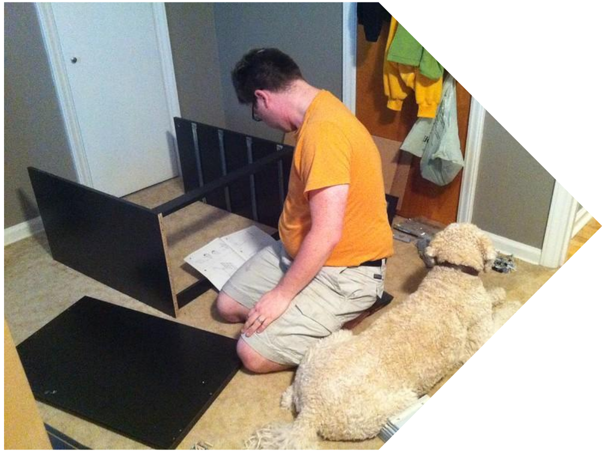
Language of reasoning in a VQA game



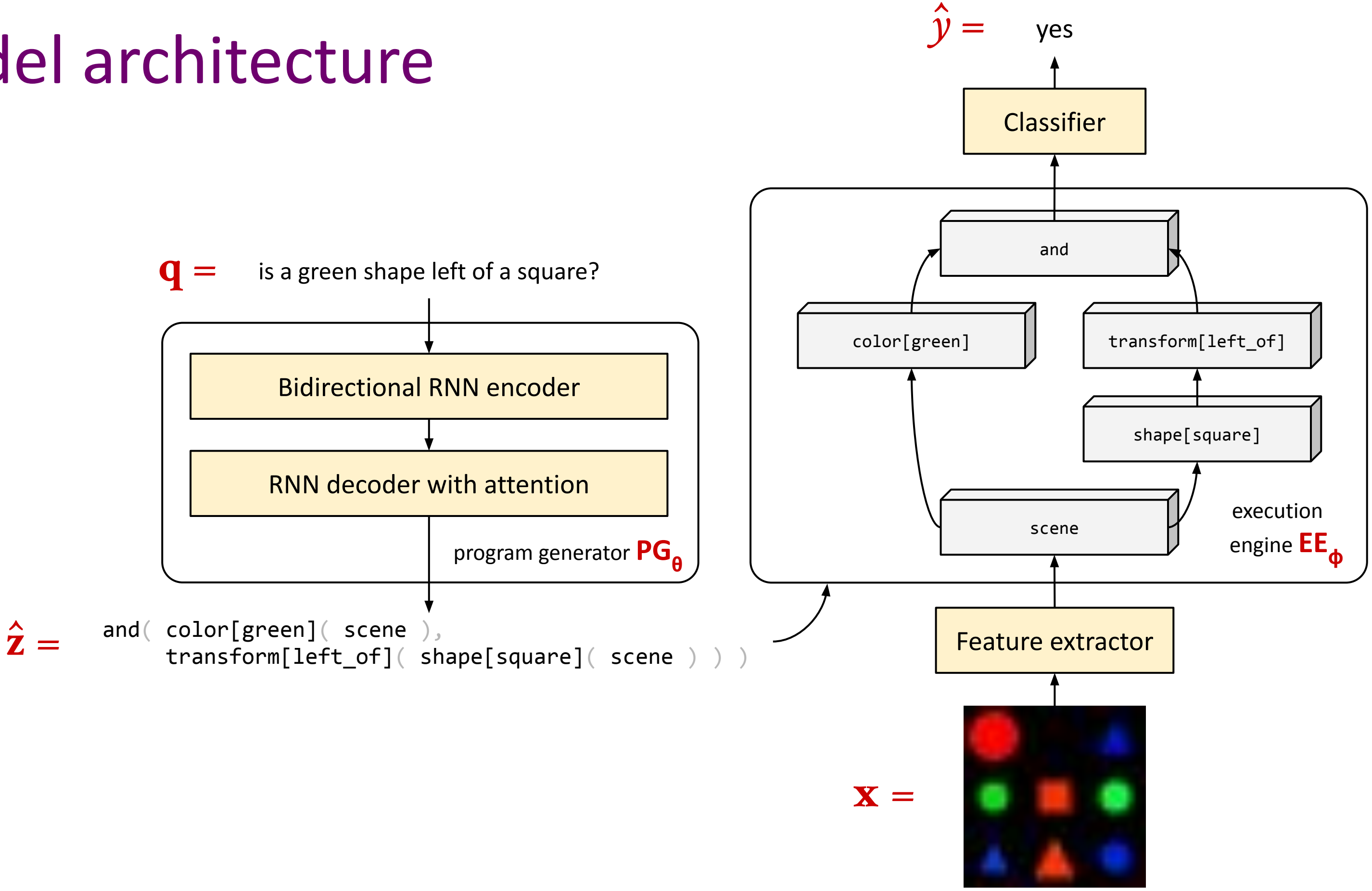
Goal: Systematic generalization

- NMNs exhibit compositionality given the *right layout*
  - NMNs can systematically generalize to SQOOP, where RelNet, FiLM, MAC fail (Bahdanau et al., 2018)
- **But**: The right layout does not emerge naturally
  - Bahdanau et al., 2018 needed to provide correct tree-structured layouts
  - Learned layouts only converged to be robust under a strong prior for the correct structure
- *Use IL to encourage structured layouts towards systematic generalization*

Method



Model architecture



Execution engine choices

	Shared module parameters	**Separate module parameters**
Tensor-valued input/output	Tensor-FiLM-NMN	Tensor-NMN
Vector-valued input/output	Vector-NMN	

Architecture from "Inferring and Executing Programs for Visual Reasoning" (Johnson et al., 2017b)

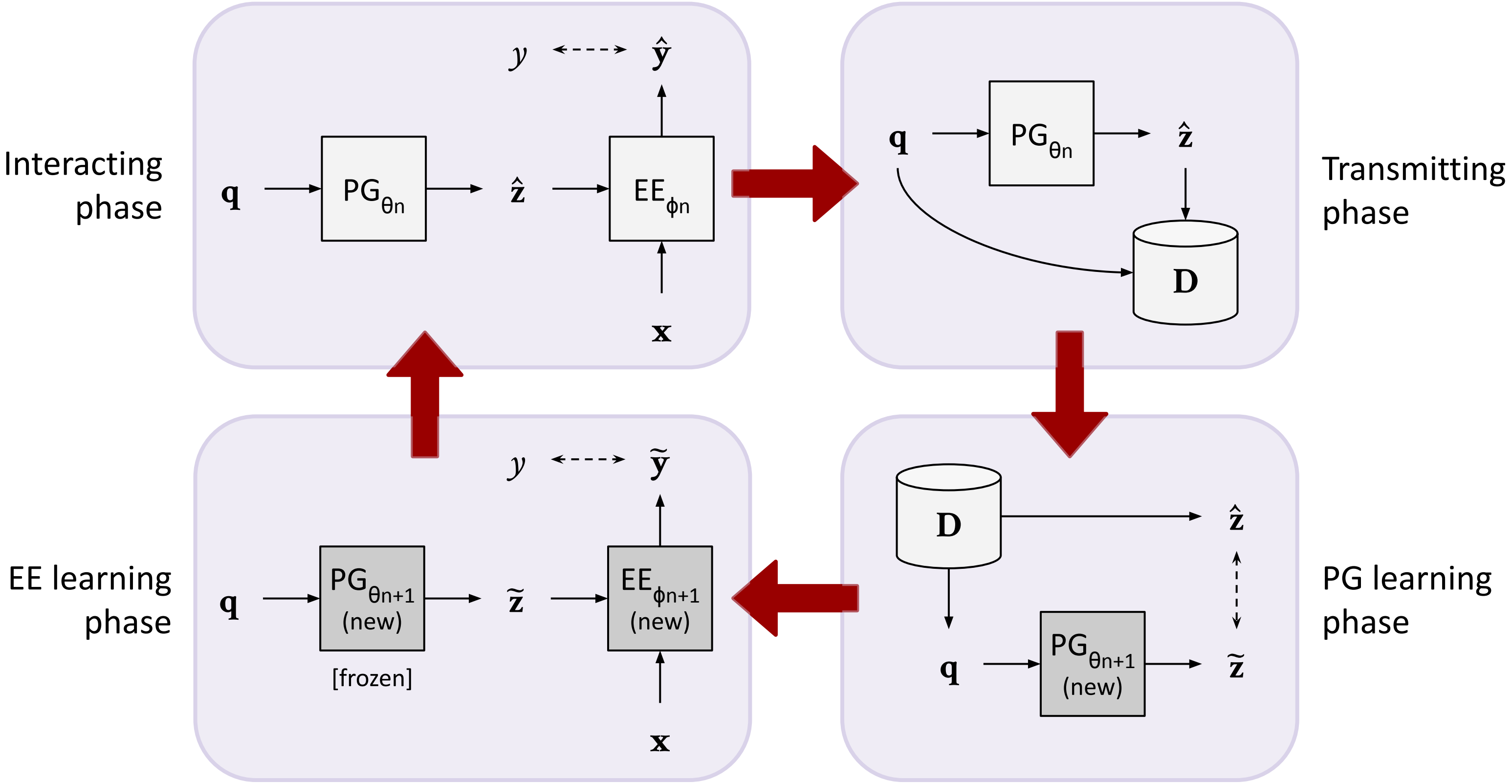
Architecture from "CLOSURE: assessing systematic generalization of CLEVR models" (Bahdanau et al., 2019)

New in this work (details in paper)

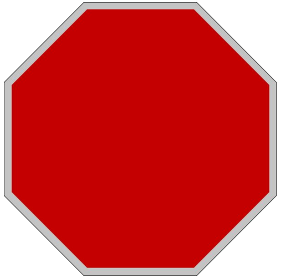
Simulating an inductive bias for structure

- Emergence of the right structure from scratch is still too hard
    - Johnson et al., 2017 needed to pre-train their model with ground-truth programs
- Use a small number of ground-truth programs for supervision
    - Supervision throughout training *simulates* an appropriate inductive bias for IL
    - More data efficient with IL than baselines

Iterated learning for NMNs



Learning bottleneck



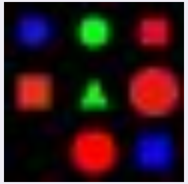
- Limit the length of the learning phases: **early stopping**
- Generally a sweet spot for the number of steps

- Program generator
  - *Too low:* Low confidence in utterances, high variance
  - *Too high:* Overfitting to transmitted data
- Execution engine
  - *Too low:* Gradients are high variance at the start of interacting phase
  - *Too high:* Overfitting to imperfect program generator

SHAPES-SyGeT (**SHAPES Sy**stematic **Ge**neralization **T**est)

Introducing a new split of the SHAPES dataset (Andreas et al., 2016)

- Evaluate systematic generalization
- Splits: Train, Val-IID, Val-OOD
  - Based on separate training and evaluation question templates



Q1 (train): Is a red shape above a green shape? — Is a *COLOR* shape *RELATIVE(1)* a *COLOR* shape?

Q2 (train): Is a *circle* left of below a *square*? — Is a *SHAPE RELATIVE(2)* a *SHAPE*?

Q3 (eval): Is a red shape left of below a green shape? — Is a *COLOR* shape *RELATIVE(2)* a *COLOR* shape?

Download from:
`https://github.com/ankitkv/SHAPES-SyGeT`

Mila

Université de Montréal

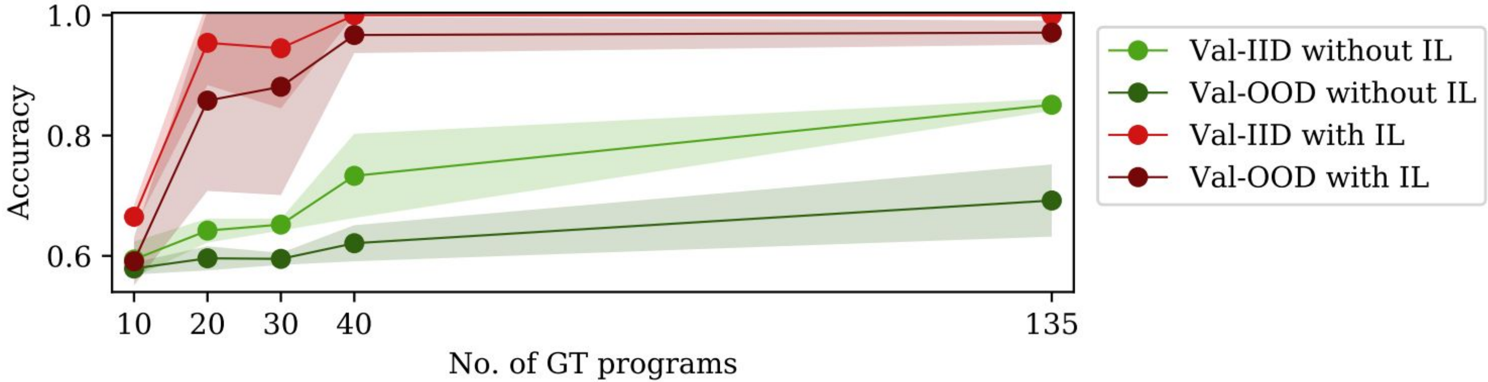
Experiments

Accuracy on SHAPES-SyGeT

Model	Val-IID		Val-OOD	
FiLM	0.720 ± 0.01		$\mathbf{0.609 \pm 0.01}$	
MAC	$\mathbf{0.730 \pm 0.01}$		0.605 ± 0.01	
	#GT 20	**#GT 135**	**#GT 20**	**#GT 135**
Tensor-NMN	0.645 ± 0.01	0.700 ± 0.01	0.616 ± 0.01	0.641 ± 0.03
Tensor-NMN+IL	0.756 ± 0.07	0.763 ± 0.04	0.648 ± 0.02	0.661 ± 0.02
Tensor-FiLM-NMN	0.649 ± 0.02	0.851 ± 0.01	0.605 ± 0.01	0.692 ± 0.06
Tensor-FiLM-NMN+IL	$\mathbf{0.954 \pm 0.07}$	$\mathbf{1.000 \pm 0.00}$	$\mathbf{0.858 \pm 0.15}$	$\mathbf{0.971 \pm 0.02}$

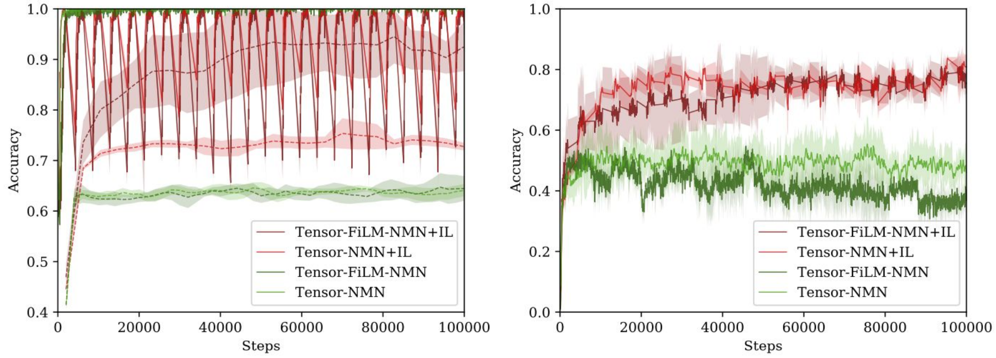
- 20 GT programs = 15%, and 135 GT programs = 100% of training programs
- IL improves generalization in Tensor-NMN and Tensor-FiLM-NMN

SHAPES-SyGeT with varying No. of GT programs



- With 40 GT programs, IL approaches the generalization performance with 135 GT programs
- Do not see the same data efficiency without IL

SHAPES-SyGeT learning curves with 20 GT programs



(a) Task accuracy. Solid lines are training and dashed lines are Val-IID.

(b) Program accuracy.

- All models achieve perfect training accuracy in 5000 steps
- Learning bottleneck encourages consistent improvement in program accuracy
    - Leads to better Val-IID and Val-OOD

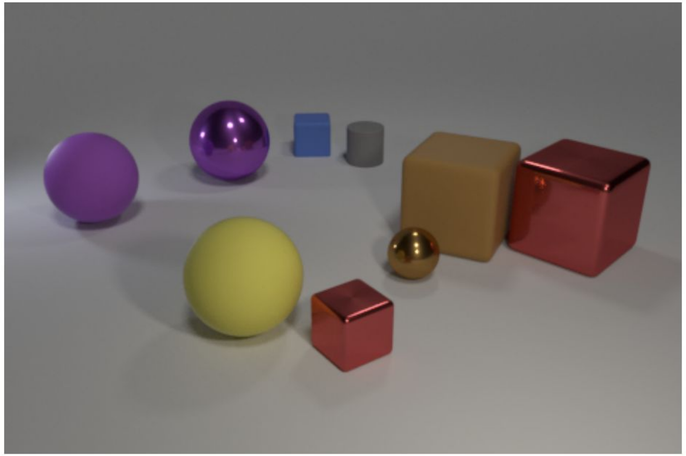
SHAPES-SyGeT example generations

Is a circle above a circle?	
Tensor-NMN :	and shape[circle] scene transform[above] transform[above] shape[circle] scene
Tensor-FiLM-NMN :	and shape[circle] scene transform[above] transform[above] shape[circle] scene
Tensor-FiLM-NMN+IL:	and shape[circle] scene transform[above] shape[circle] scene
Is a blue shape above a square?	
Tensor-NMN :	and color[blue] scene transform[above] transform[above] shape[square] scene
Tensor-FiLM-NMN :	and color[blue] scene transform[above] shape[square] scene
Tensor-FiLM-NMN+IL:	and color[blue] scene transform[above] shape[square] scene
Is a green shape red?	
Tensor-NMN :	and color[green] scene color[red] scene
Tensor-FiLM-NMN :	and color[green] scene color[green] scene
Tensor-FiLM-NMN+IL:	and color[green] color[red] scene
Is a green shape left of below a red shape?	
Tensor-NMN :	and color[green] scene transform[left_of] color[red] scene
Tensor-FiLM-NMN :	and color[green] scene transform[left_of] color[red] scene
Tensor-FiLM-NMN+IL:	and color[green] scene transform[left_of] transform[below] color[red] scene
Is a green shape below above a circle?	
Tensor-NMN :	and color[green] scene transform[below] shape[circle] scene
Tensor-FiLM-NMN :	and color[green] scene transform[above] transform[below] shape[circle] scene
Tensor-FiLM-NMN+IL:	and color[green] scene transform[below] transform[above] shape[circle] scene

Particularly interesting failures to systematically generalize

CLEVR/CLOSURE example

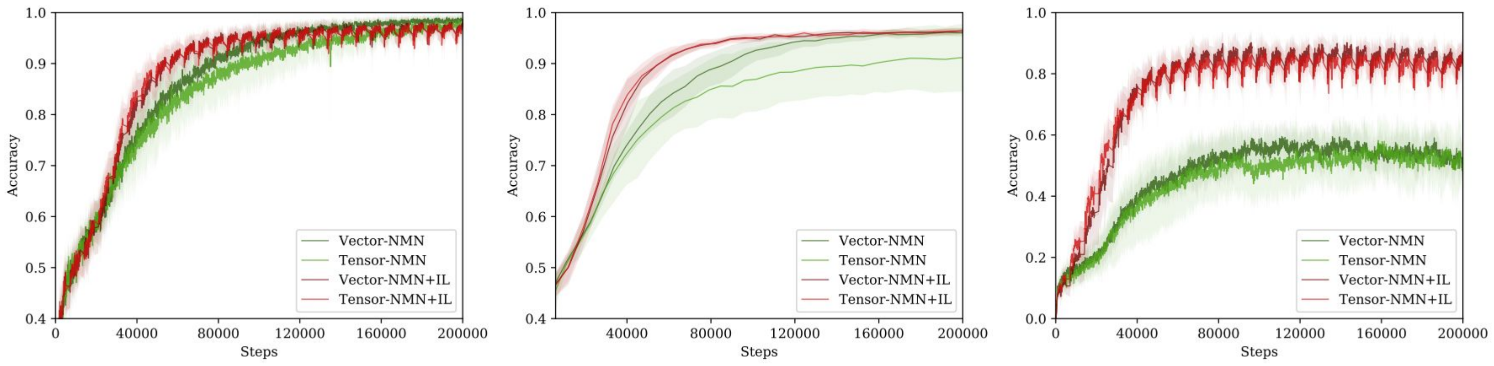
CLEVR (Johnson et al., 2017)
CLOSURE (Bahdanau et al., 2019)



Q1 (CLEVR): There is another cube that is the same size as the brown cube; what is its color?
Q2 (CLEVR): There is a thing that is in front of the yellow thing; does it have the same color as cylinder?
Q3 (CLOSURE): There is another rubber object that is the same size as the gray cylinder; does it have the same color as the tiny shiny block?

(Image from Bahdanau et al., 2019)

CLEVR/CLOSURE learning curves with 100 GT programs



(a) Task training accuracy. (b) Task validation accuracy. (c) Program accuracy.

- Tensor-FiLM-NMN performs slightly worse than Vector-NMN, ignore for simplicity
- All models reach similar training accuracies
- IL leads to significantly higher program accuracy
- The generalization difference is more apparent in the OOD CLOSURE dataset

Mila

Université de Montréal

Accuracy on CLOSURE categories with 100 GT programs

Evaluation set	Tensor-NMN		Vector-NMN	
	Without IL	With IL	Without IL	With IL
CLEVR-Val	0.912 ± 0.07	$\mathbf{0.964 \pm 0.01}$	0.960 ± 0.01	$\mathbf{0.964 \pm 0.00}$
and_mat_spa	$\mathbf{0.278 \pm 0.17}$	0.264 ± 0.16	$\mathbf{0.400 \pm 0.13}$	0.335 ± 0.18
or_mat	0.327 ± 0.11	$\mathbf{0.481 \pm 0.24}$	0.367 ± 0.11	$\mathbf{0.563 \pm 0.23}$
or_mat_spa	0.286 ± 0.13	$\mathbf{0.405 \pm 0.22}$	0.330 ± 0.11	$\mathbf{0.444 \pm 0.24}$
compare_mat	0.793 ± 0.11	$\mathbf{0.851 \pm 0.17}$	0.660 ± 0.16	$\mathbf{0.873 \pm 0.12}$
compare_mat_spa	0.746 ± 0.13	$\mathbf{0.853 \pm 0.15}$	0.677 ± 0.14	$\mathbf{0.871 \pm 0.12}$
embed_spa_mat	0.824 ± 0.07	$\mathbf{0.947 \pm 0.03}$	0.863 ± 0.07	$\mathbf{0.900 \pm 0.08}$
embed_mat_spa	0.739 ± 0.14	$\mathbf{0.941 \pm 0.02}$	0.894 ± 0.03	$\mathbf{0.936 \pm 0.03}$

- IL improves performance on all but one CLOSURE category
- Tensor-NMN with IL leads to CLEVR accuracy similar to previous works with far fewer programs
  - 18000 for Johnson et al. (2017b), 1000 for Vedantam et al. (2019)

GQA example

GQA (Hudson & Manning, 2019)

(Image from Hudson & Manning, 2019)

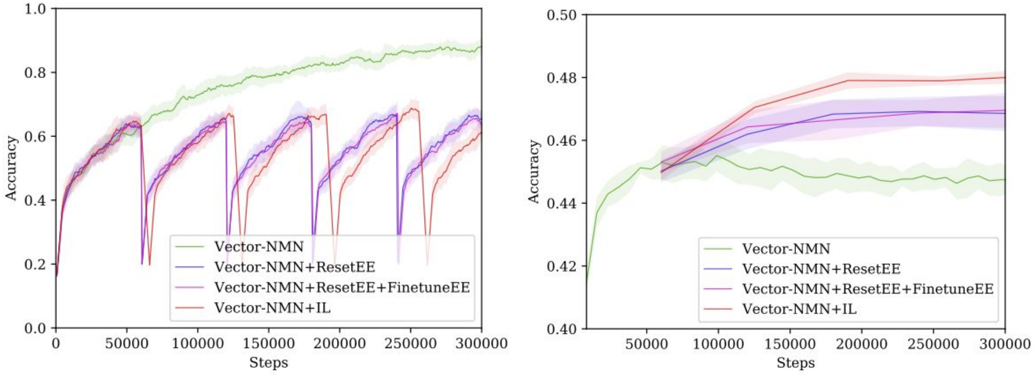
Q1: Is the tray on top of the table black or light brown?
Q2: Is the small table both oval and wooden?
Q3: Is there any fruit to the left of the tray the cup is on top of?

Accuracy on GQA

#GT programs	Model	Accuracy
-	`Vector-NMN` on GT programs	$\mathbf{0.556 \pm 0.002}$
943000	`Vector-NMN`	$\mathbf{0.486 \pm 0.002}$
4000	`Vector-NMN`	0.455 ± 0.006
4000	`Vector-NMN`+ResetEE	0.469 ± 0.007
4000	`Vector-NMN`+ResetEE+FinetuneEE	0.470 ± 0.007
4000	`Vector-NMN`+IL	$\mathbf{0.480 \pm 0.003}$

- Huge program vocabulary, cannot use Tensor-NMN
- Execution engine tends to overfit the training images
  - Stronger baselines: Partial IL: reset the EE while retaining old FiLM embeddings
- Full IL algorithm is necessary for best generalization
  - Approaching accuracy of using all GT programs by using only 0.4%

GQA learning curves with 4000 GT programs



(a) Task training accuracy.

(b) Task validation accuracy.

- ResetEE baselines and IL have similar training curves, but IL has distinctly better generalization
  - Due to better emergent program structure

GQA program accuracies

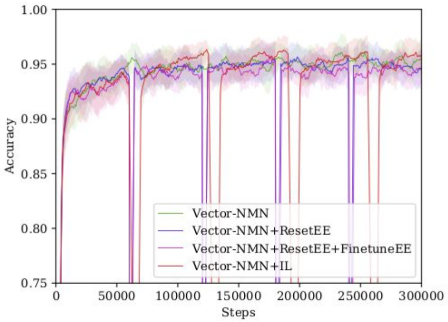
Op: Operation

Subop: Sub-operation

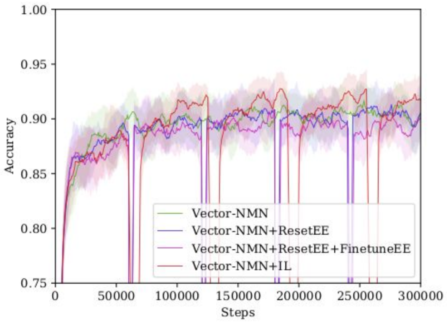
Arg: Argument

ArgNeg: Argument negation

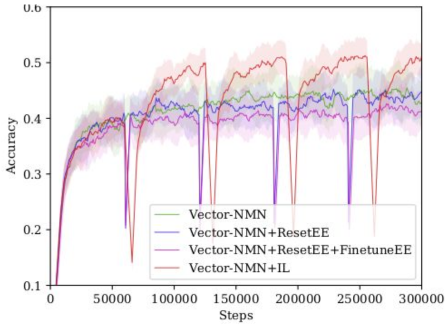
Arity: Number of operands



(a) Op.



(b) Op+Subop.

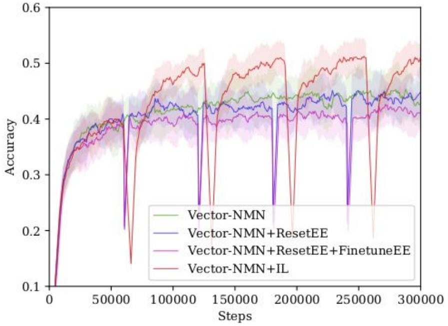


(c) Op+Subop+Arg.

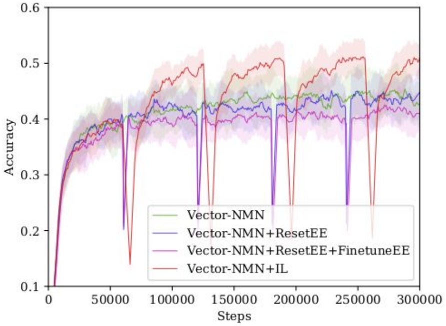
Q: does the lady to the left of the player wear a skirt?

Program:

```
verify.rel(skirt,wearing,o){1}
relate(lady,to the left of,s){1}
select(player){0}
```



(d) Op+Subop+Arg+ArgNeg.



(e) Op+Subop+Arg+ArgNeg+Arity.

Mila

Université de Montréal

Summarizing

Conclusion

- We demonstrated that IL aids learning of ground-truth program structure
  - SHAPES-SyGeT: toy
  - CLEVR/CLOSURE: intermediate
  - GQA: heavy-duty
- Program generators with correct structure systematically generalize better
- IL deserves to be explored in broad range of machine learning applications

References (1/2)

Simon Kirby, Tom Griffiths, and Kenny Smith. "Iterated learning and the evolution of language." In: Current opinion in neurobiology 28 (2014), pp. 108–114.

Simon Kirby, Hannah Cornish, and Kenny Smith. "Cumulative cultural evolution in the laboratory: An experimental approach to the origins of structure in human language." In: Proceedings of the National Academy of Sciences 105.31 (2008), pp. 10681–10686.

Shangmin Guo, Yi Ren, Serhii Havrylov, Stella Frank, Ivan Titov, and Kenny Smith. "The emergence of compositional languages for numeric concepts through iterated learning in neural agents." In: arXiv preprint arXiv:1910.05291 (2019).

Fushan Li and Michael Bowling. "Ease-of-teaching and language structure from emergent communication." In: Advances in Neural Information Processing Systems. 2019, pp. 15851–15861.

Michael Cogswell, Jiasen Lu, Stefan Lee, Devi Parikh, and Dhruv Batra. "Emergence of compositional language with deep generational transmission." In: arXiv preprint arXiv:1904.09067 (2019).

Gautier Dagan, Dieuwke Hupkes, and Elia Bruni. "Co-evolution of language and agents in referential games." In: arXiv preprint arXiv:2001.03361 (2020).

Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B Cohen, and Simon Kirby. "Compositional languages emerge in a neural iterated learning model." In: arXiv preprint arXiv:2002.01365 (2020).

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. "CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning." In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017, pp. 2901–2910.

References (2/2)

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. "Systematic generalization: what is required and can it be learned?" In: arXiv preprint arXiv:1811.12889 (2018).

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. "Neural module networks." In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 39–48.

Dzmitry Bahdanau, Harm de Vries, Timothy J O'Donnell, Shikhar Murty, Philippe Beaudoin, Yoshua Bengio, and Aaron Courville. "CLOSURE: Assessing Systematic Generalization of CLEVR Models." In: arXiv preprint arXiv:1912.05783 (2019).

Drew A Hudson and Christopher D Manning. "GQA: a new dataset for real-world visual reasoning and compositional question answering." In: Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. "Inferring and executing programs for visual reasoning." In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2989–2998, 2017b.

Ramakrishna Vedantam, Karan Desai, Stefan Lee, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. "Probabilistic neural symbolic models for interpretable visual question answering." In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, pp. 6428–6437. PMLR, 2019.

Let’s discuss!

Full paper

For further questions, reach out at
`ankit.vani@umontreal.ca`