Supplementary Material

Interpretable Visual Reasoning via Probabilistic Formulation under Natural Supervision

Xinzhe Han et.al

1 Implement Details for Concrete Distribution

The Gumbel-softmax trick [4] is an attempt to overcome the inability to apply the re-parameterization trick to discrete data, which is widely used in generative models like VAEs [7] and GANs [3]. According to [4], samples from Concrete distribution with parameter $\tau \in (0, \infty)$, $\pi_k \in (0, \infty)$ are

$$x_k = \frac{\exp\left(\left(\log \pi_k + G_k\right)/\tau\right)}{\sum_{i=1}^{n} \exp\left(\left(\log \pi_i + G_i\right)/\tau\right)} \tag{1}$$

where G_k is i.i.d sampled from Gumbel(0,1)[1]. The log-density of $\mathcal{C}(\pi, \tau)$ is computed as

$$\log p_{\pi,\lambda}(X) = \log \Gamma(n) + (n-1)\log\tau + \sum_{i=1}^{k} \log \frac{\pi_k x_k^{-\tau-1}}{\sum_{i=1}^{n} \pi_i x_i^{-\lambda}} \tag{2}$$

Let $\alpha_k = \log \pi_k$, $\alpha_k \in (-\infty, +\infty)$ can be parametrized without constrains. Eq. 2 can be written as

$$\log p_{\pi,\tau}(X) = \log \Gamma(n) + (n-1)\log\tau + \sum_{i=1}^{k} \{\log[\mathrm{Softmax}(\alpha_k - \tau \log x_k)] - \log(x_k)\} \tag{3}$$

Since $x_k \in [0, 1]$, we find that the scale of $\log x_k$ is quite unstable, making the network hard to converge. Note that the K-L terms of a variational loss are invariant under invertible transformation, we can reduce the variance by directly sampling $y_k = \log x_k$.

$$y_k = \frac{\alpha_k + G_k}{\tau} - \log \sum_{i=1}^{n} \exp\left\{\frac{\alpha_i + G_i}{\tau}\right\} \tag{4}$$

It is obvious that $\exp(y_k) \sim \mathrm{Concret}(\pi_k, \tau)$. The log-probability of Y can be transformed to

$$\log \kappa_{\pi,\tau}(Y) = \log \Gamma(n) + (n-1)\log\tau + \sum_{i=1}^{k} \{\log[\mathrm{Softmax}(\alpha_k - \tau y_k)]\} \tag{5}$$

We can avoid calculating the unstable term $\log x_k$, and the K-L divergence keeps invariant. The real samples are $x_k = \exp(y_k)$.

[1] The Gumbel(0, 1) distribution can be sampled using inverse transform sampling by drawing $u_k \sim \mathrm{Uniform}(0, 1)$ and computing $G_k = -\log(-\log(u_k))$

2 Implementation Details for TRN

We implement TRN as an injected module to both a classical one-stage method Bottom-up Top-down Attention (UpDn) [1] and a widely used implicit reasoning method Bilinear Attention Network (BAN) [6]. To verify the function of fusion strategy, we further inject TRN to DFAF in experiments on CLEVR.

UpDn is a representative method with only singe time-step State Inference. So we add T Trans Cells and Entity Cells to realize temporal reasoning. Trans cell that assumes a simple one-step Markov transition process from the current state to the next state. The transition matrix $\mathbf{T}_t$ is defined as attention on the edge based on transition phrase h_r^t.

$$\begin{aligned} \mathbf{T}_t &= \underset{c}{\text{Softmax}}[L_3(L_1(\mathbf{h}_r^t) \odot L_2(\mathbf{v}_r))] \\ \pi_p^{t+1} &= \mathbf{T}_t \pi_q^t \end{aligned} \tag{6}$$

where L_1, L_2 and L_3 are linear mappings, $\odot$ is Hadamard product, $\mathbf{h}_r^t \in \mathbb{R}^{K \times K \times d_q}$ is the transition phrase embedding expanded from h_e^t, and Softmax_c denotes normalization operation along column direction.

On the other hand, the Entity Cell takes the linear attention between h_e^t and node features $\mathbf{v}_n$ as the posterior distribution parameter $\pi_q^t \in \mathbb{R}^K$

$$\pi_q^t = L_6(L_4(\mathbf{h}_e^t) \odot L_5(\mathbf{v}_n)) \tag{7}$$

where L_4, L_5 and L_6 are linear mappings of feature vectors. The final state output is the matrix product of sample $\tilde{z}_T$ from $q(z_T|h_e^t, \mathbf{v})$ and node vector

$$b_T = \tilde{z}_T \mathbf{v}_n \tag{8}$$

In order to reduce computational cost on train stage, we simply stack reasoning blocks with shared parameters.

As for BAN, stacked attention can be viewed as multiple-step State Inference, but it lacks time-dependent State Transition. For Entity Cells, we first substitute the $L \times q_d$ question embeddings for bilinear attention with $T \times q_d$ entity embeddings. We directly sum up the bilinear fusion matrix before Softmax function $\mathcal{A} \in \mathbb{R}^{K \times T}$ along row direction as the State Inference posterior distribution parameter:

$$\pi_q^t = \sum_j \mathcal{A}_{i,j} \tag{9}$$

The time dependency between reasoning blocks is modelled by Trans Cells the same as Eq. 6. The output feature b_T and answer classification model remain unchanged as the original BAN.

For DFAF, inter-modal attention infers the latent states aggregating information from both questions and images, which can be regarded as State Inference terms. Similar to BAN, Entity Cells calculate the parameter π_q as

$$\pi_q^t = \sum_c \mathcal{A}^t = \sum_c R_r^t L_6(\mathbf{h}_e^{tT}) \tag{10}$$

Algorithm 1: Temporal Reasoning Model

Input: Node features $\mathbf{v}_n$, Relation features $\mathbf{v}_r$, Quetion words $\mathbf{x}$
Initialize: $h_e^{1:T}, h_r^{1:T} \leftarrow \text{BiGRU}(\mathbf{x})$, , $q(z_0) \leftarrow \mathcal{C}(\mathbf{0})$;
for $t = 1 \dots T$ **do**
  $p(z_{t-1}) \leftarrow q(z_{t-1})$
  $\pi_q^t \leftarrow f_1(\mathbf{v}_n, h_e^t),\ q(z_t) \leftarrow \mathcal{C}(\pi_q^t)$
  $\pi_b^t \leftarrow f_2(\mathbf{v}_n, h_e^t, \pi_t),\ q(z_{t-1}) \leftarrow \mathcal{C}(\pi_b^t)$
  $\pi_p^{t+1} = f_3(h_r^t, \mathbf{v}_r, \pi_q^t),\ p(z_{t+1}) \leftarrow \mathcal{C}(\pi_b^t)$
  Sample $\tilde{z}_t \sim q(z_t)$, $\tilde{z}_{t-1} \sim q(z_{t-1})$
  $b_t = \mathbf{v}_n \times \tilde{z}_t$
  $\tilde{x}_t \leftarrow \text{MLP}(b_t)$
  $KL_{z_t} \leftarrow \log q(\tilde{z}_t) - \log p(\tilde{z}_t)\ KL_{z_{t-1}} \leftarrow \log q(\tilde{z}_{t-1}) - \log p(\tilde{z}_{t-1})$
  $p(x_t) \leftarrow \text{BCE}(x, x_t),\ \mathcal{L}_t = p(x_t) - KL_{z_t} - KL_{z_{t-1}}$
  return $\mathcal{L}_t$, b_t, $q(z_t)$, $p(z_{t+1})$
end
$\tilde{a} \leftarrow \text{MLP}(b_T, h_e^T)$
$\mathcal{L}(\Theta) \leftarrow -\text{BCE}(a, \tilde{a}) - \sum_{t=1}^{T} \mathcal{L}_t$
Update model parameters $\Theta \leftarrow \Theta - \alpha \nabla_\Theta \mathcal{L}(\Theta)$

where $R_r^t \in \mathbb{R}^{K \times d}$ is the embedding for node features at time step t, L_7 denotes a linear transformation that projects $\mathbf{h}_e^{tT}$ to the dimension of node features.

Different from BAN, the DyIntraMAFR$_{R \leftarrow R}$ in DFAF performs intra-modal information transformation, which is similar to State Transition on fully-connected question conditioned graph. Therefore, in order to remain original fusion strategies and obtain the time-dependency between cells, Trans Cells replace conditional gates with transition phrase embedding $\mathbf{h}_r^t$ at the current state. $\mathbf{T}_t$ can be calculated as

$$\begin{aligned} G_{R \leftarrow E}^t &= \sigma\left(L_7\left(\mathbf{h}_r^t\right)\right) \\ \hat{R}^t &= \left(1 + G_{R \leftarrow E}\right) \odot R^t \\ \mathbf{T}_t &= \text{Softmax}\left(\frac{\hat{R}^t \hat{R}^{tT}}{\sqrt{dim}}\right) \end{aligned} \tag{11}$$

where L_7 indicates the linear transformation, and dim is the dimension of node features. With the help of additional Generative Reconstruction and State Transition, we can improve the performance and interpretation compared with original DFAF.

The computation procedure of the whole network is presented in Algorithm 1.

3 Experiment Settings

We evaluate our model on both real-world dataset VQA v2.0 [2] and synthetic dataset CLEVR [5] without using the labelled programs.

For VQA v2, the temperature of Concrete distribution is set to be 2.5, the maximum number of entities (the number of Temporal Reasoning Blocks) is set

as 3. All models are trained with Adamax optimizer. The batch size is set as 64. The learning rate is set as 2e-3 with warm-up strategy for the first 3 epochs. All initializations are Pytorch default initialization.

For CLEVR, we trained on CLEVR train split and test on validation split. All models are trained with Adamax optimizer. The temperature of Concrete distribution is set to be 2.0, the maximum number of entities (the number of Temporal Reasoning Blocks) is set as 5. The batch size is set as 128. The learning rate is set as 1e-3 with warm-up strategy for the first 3 epochs. For CLEVR-Humans, we first pre-train our model on CLEVR train split, and then fine-tune on CLEVR-Humans train split with learning rate of 1e-4. All initializations are Pytorch default initialization.

4 Additional Experiments

4.1 Design Choice

The number TRN blocks. We set the number of blocks for VQA/CLEVR as 3/5 because the number of entity phrases in most questions is no more than 2/4 (with 1 global embedding). Adding more blocks does not increase its effectiveness, KL divergence and reconstruction loss for blocks after the last entity embedding will be masked. As shown in Table 1, extra blocks may confuse the final decision and visualizations (Exp.3 in Fig.4 , the last state becomes meaningless).

Table 1. Ablation study for the number of TRN blocks

# Blocks	VQA v2 val				CLEVR val			
	2	3	4	5	3	4	5	6
UpDn+TRN	63.64	64.12	64.13	64.10	77.83	86.14	88.67	83.98
BAN+TRN	65.27	65.31	65.34	65.41	80.6	82.19	85.24	81.96
BAN	65.11	65.17	65.21	65.45	74.7	76.64	82.08	80.62

Latent distribution. Random variables of Concrete distribution are similar to Softmax attention. Therefore, it is chosen for fair comparison with attention-base baselines and avoids extra fusion strategy.

4.2 Quantitative Evaluation for Interpretability

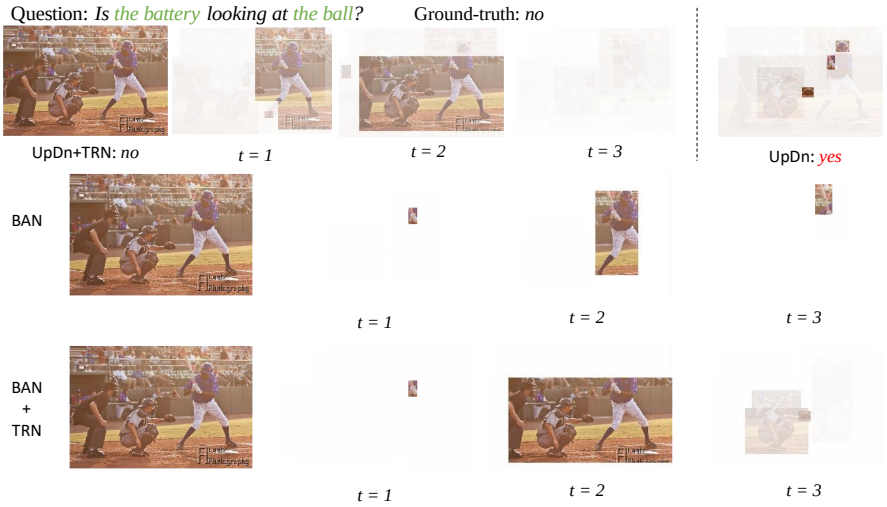
Level of interpretability is too subjective. it is too difficult to quantitatively compare the visual vector with functional program in language. Apart form qualitative comparison with labelled program (Figure 4), we also design a *Subjective Blind Test* for more convincing analysis.

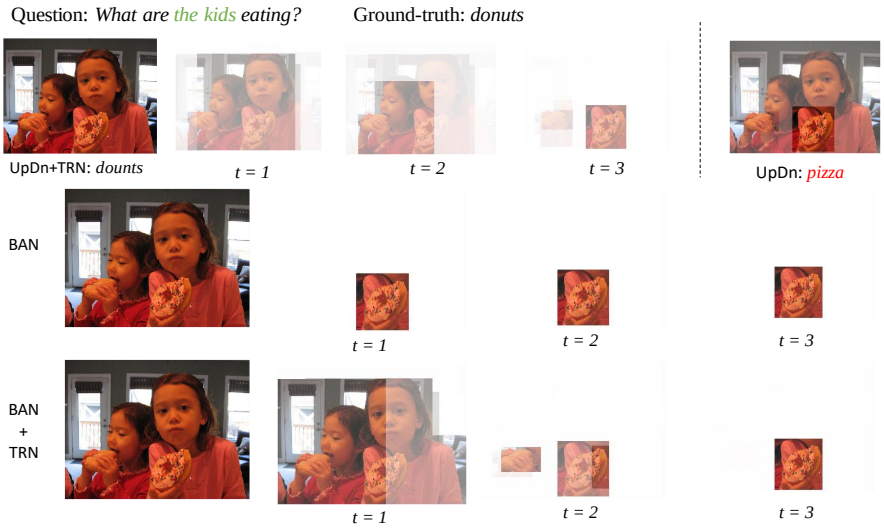
We randomly select 15 groups of visualization from UpDn+TRN, BAN and BAN+TRN (hiding the label of methods). There are 6 level for rating:

- 0: Totally wrong evidence
- 1: Noisy answer evidence or reasoning evidence, no reasoning process
- 2: Clear answer evidence, no reasoning evidence
- 3: Clear answer evidence, noisy reasoning evidence
- 4: Clear answer evidence and reasoning evidence, somehow understandable reasoning process
- 5: Clear evidence and completely understandable reasoning process

We recruit 43 amateurs with no background knowledge on VQA for rating. The mean score and standard dev. of BAN / UpDn+TRN / BAN+TRN are 1.61±0.70 / 3.20±0.94 / 3.37±0.94. Most participants believe TRN can find more clear evidence and understandable reasoning process, while BAN only locates answer-related objects.

5 Qualitative Evaluation for TRN




Question: What are the kids eating?
Ground-truth: donuts
UpDn+TRN: dounts
t = 1
t = 2
t = 3
UpDn: pizza
BAN
t = 1
t = 2
t = 3
BAN + TRN
t = 1
t = 2
t = 3


Question: what is the man wearing on his head?
Ground-truth: hat
UpDn+TRN: hat
t = 1
t = 2
t = 3
UpDn: hat
BAN
t = 1
t = 2
t = 3
BAN + TRN
t = 1
t = 2
t = 3

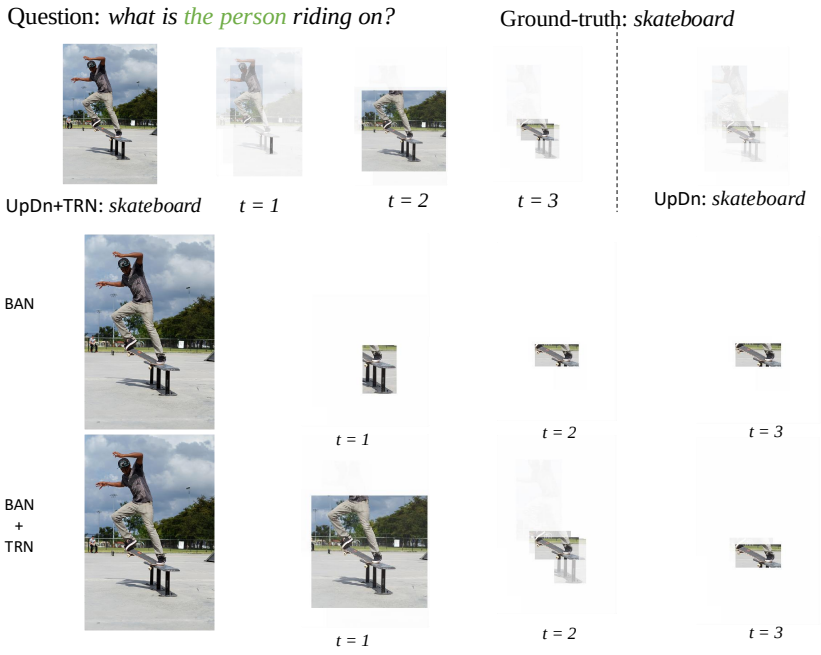


Fig. 1. Extra Examples From VQA v2. The first row is the comparison between UpDn and UpDn+TRN. TRN can display the reasoning process and improve performance without extra computational cost in the testing stage. Examples in the lower rows are the comparison between BAN and BAN+TRN. TRN can boost the interpretation with the reasoning process closer to human understanding.

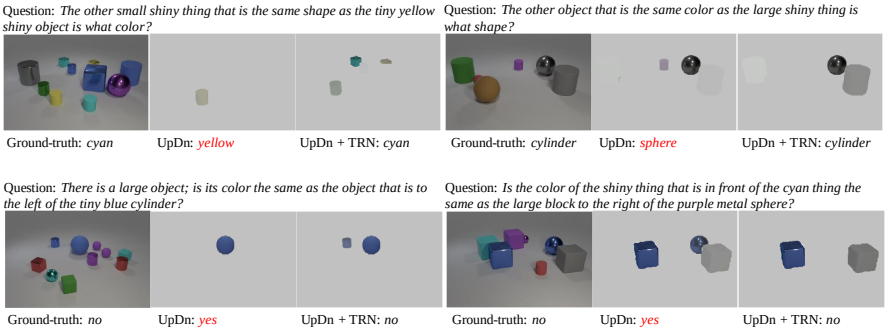


Fig. 2. More comparisons between UpDn and UpDn+TRN. TRN can offer right answers for questions that fail to be answered by UpDn.

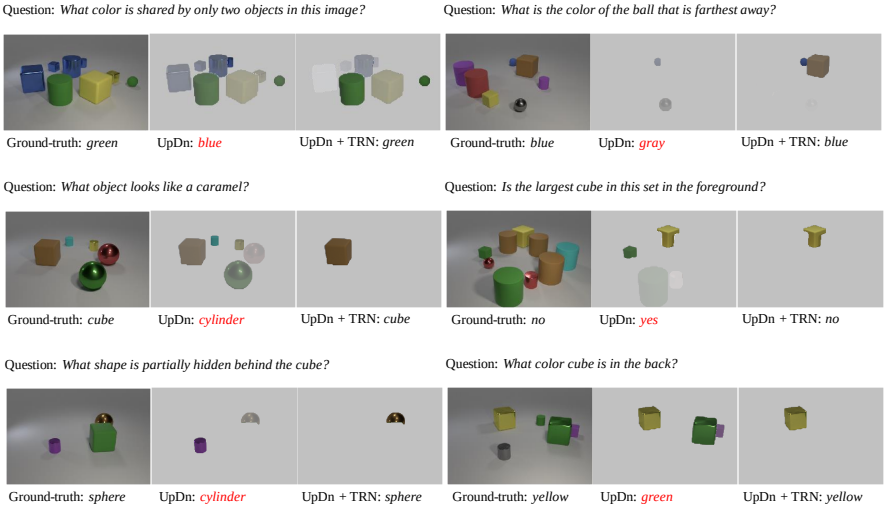
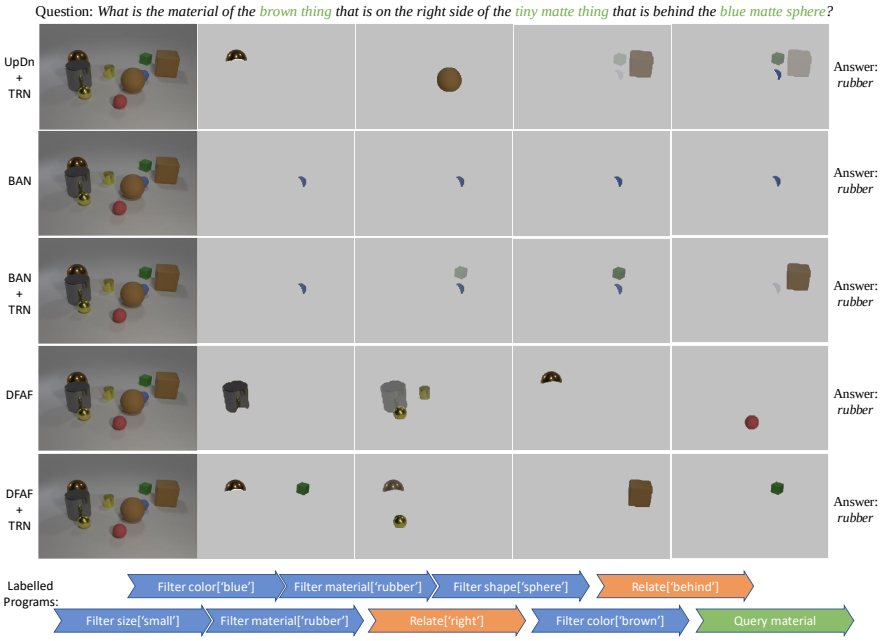
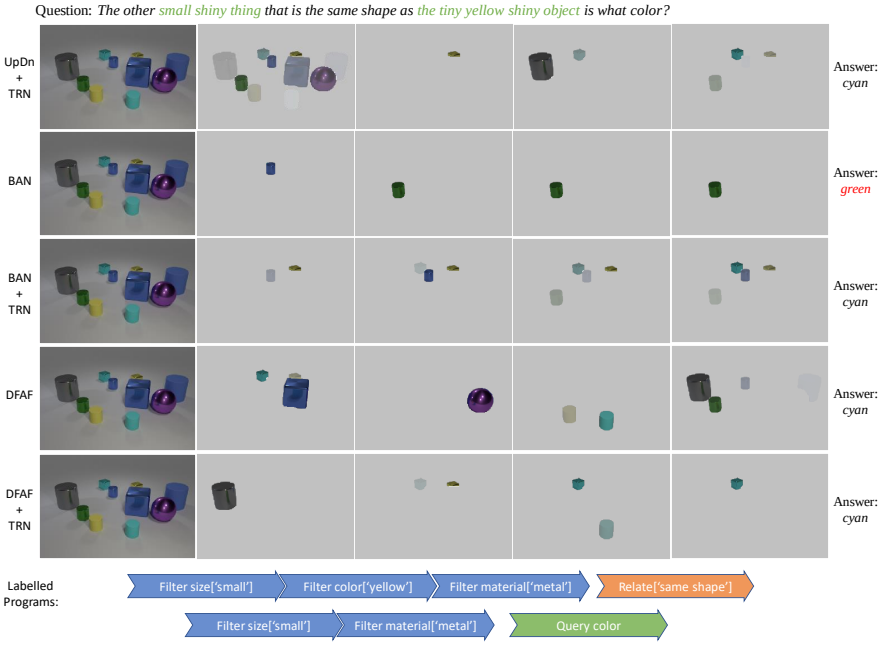


Fig. 3. Comparisons between UpDn and UpDn+TRN on CLEVR-Humans.


Question: What is the material of the brown thing that is on the right side of the tiny matte thing that is behind the blue matte sphere?
UpDn + TRN
Answer: rubber
BAN
Answer: rubber
BAN + TRN
Answer: rubber
DFAF
Answer: rubber
DFAF + TRN
Answer: rubber
Labelled Programs:
Filter color['blue'] Filter material['rubber'] Filter shape['sphere'] Relate['behind']
Filter size['small'] Filter material['rubber'] Relate['right'] Filter color['brown'] Query material


Question: The other small shiny thing that is the same shape as the tiny yellow shiny object is what color?
UpDn + TRN
Answer: cyan
BAN
Answer: green
BAN + TRN
Answer: cyan
DFAF
Answer: cyan
DFAF + TRN
Answer: cyan
Labelled Programs:
Filter size['small'] Filter color['yellow'] Filter material['metal'] Relate['same shape']
Filter size['small'] Filter material['metal'] Query color

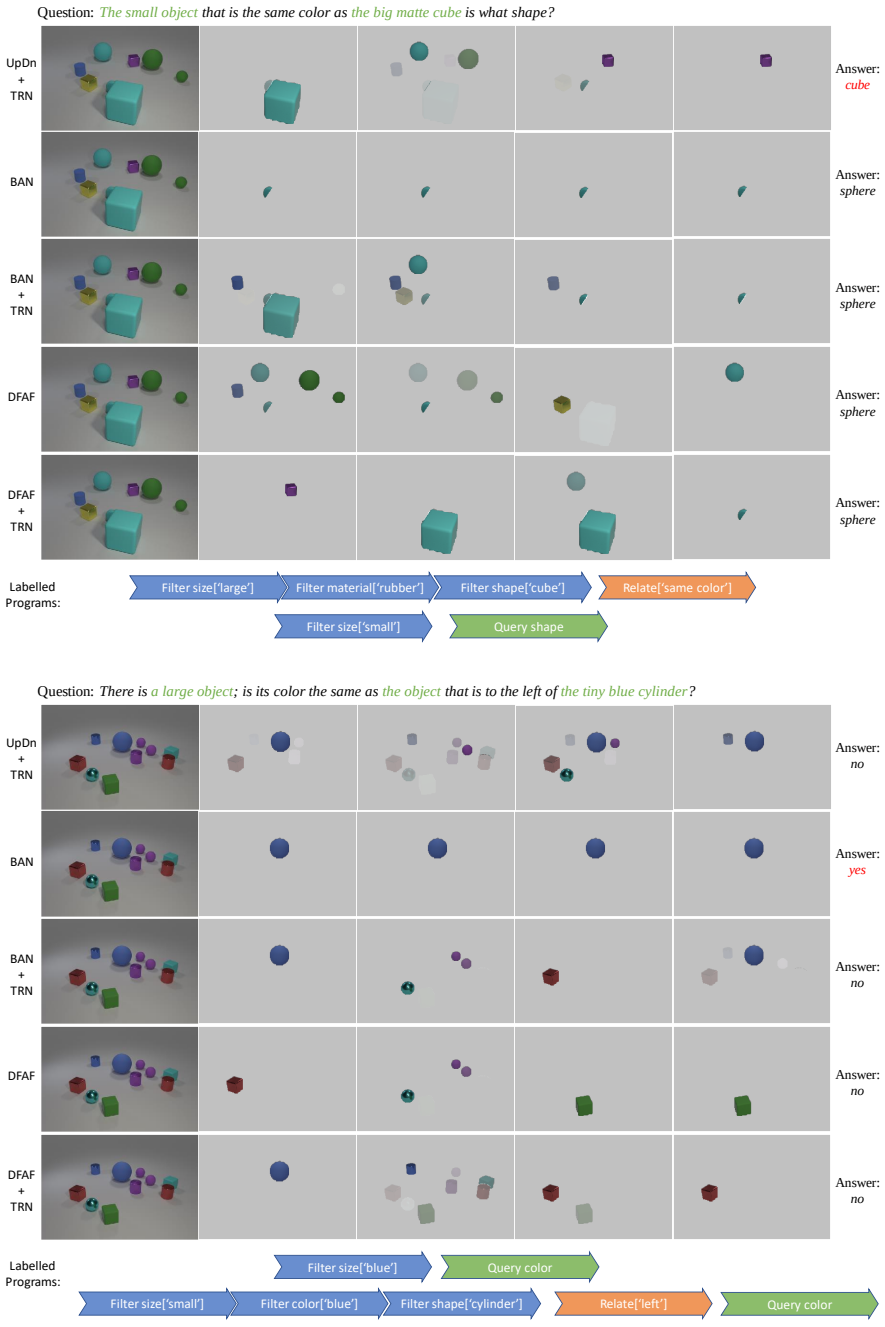


Fig. 4. Examples for visualizing reasoning process on CLEVR. TRN is closer to labelled programs and human understanding.

6 Failed Cases

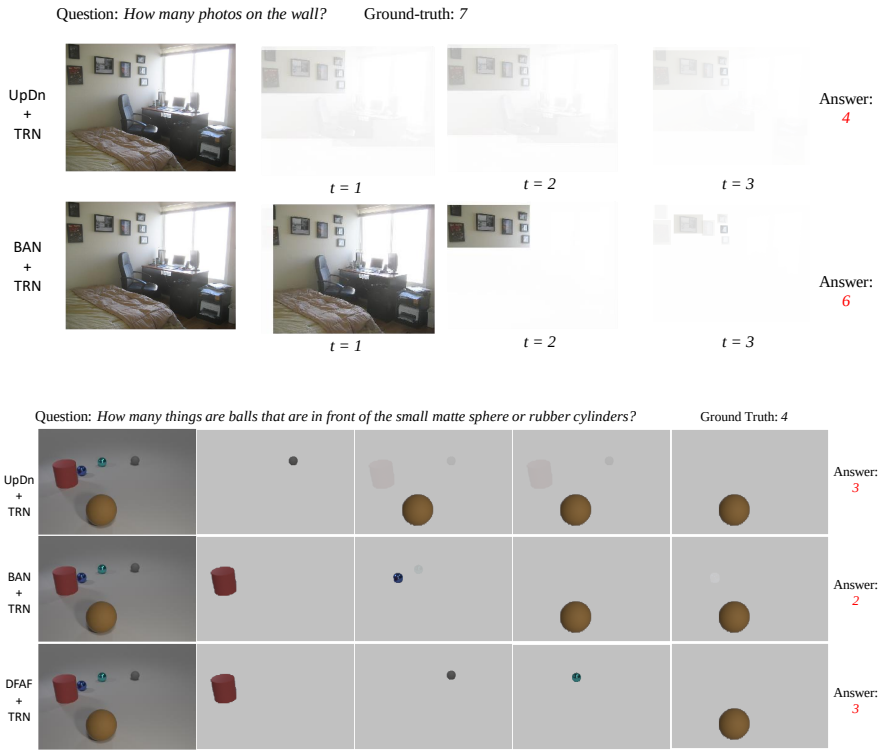


Fig. 5. Failed cases for counting problems. This is a common challenge for attention-based methods

Fig. 6. Failed case for problems that need common senses. TRN forces the model to ground information in images, but common senses cannot be inferred with traditional visual reasoning methods. Visual Common Sense Reasoning (VCR) [8] is another area that needs investigating in the future.

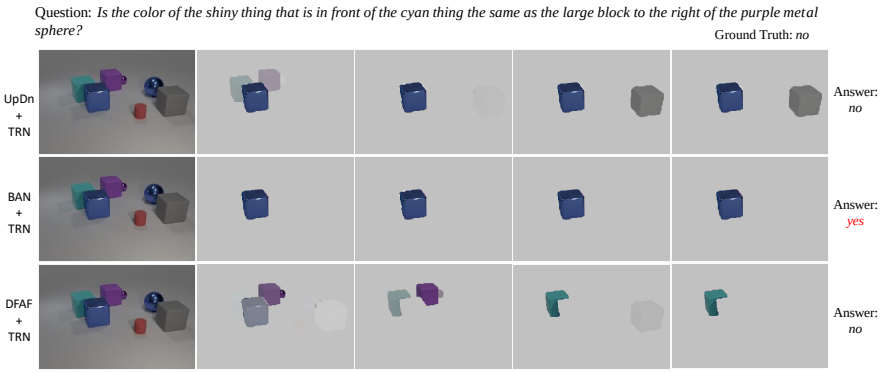


Fig. 7. Failed case for adverbial problem in complex questions. The performance could be improved by fusion strategy like DFAF, but is still poor in the reasoning process visualization.

References

1. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6077–6086 (2018) 2
2. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: Vqa: Visual question answering. In: Proceedings of the IEEE international conference on computer vision. pp. 2425–2433 (2015) 3
3. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014) 1
4. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016) 1
5. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2901–2910 (2017) 3
6. Kim, J.H., Jun, J., Zhang, B.T.: Bilinear attention networks. In: Advances in Neural Information Processing Systems. pp. 1564–1574 (2018) 2
7. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013) 1
8. Zellers, R., Bisk, Y., Farhadi, A., Choi, Y.: From recognition to cognition: Visual commonsense reasoning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019) 12