Interpretable Visual Reasoning via Probabilistic Formulation under Natural Supervision

Xinzhe Han[1,2], Shuhui Wang[2] (✉), Chi Su[3], Weigang Zhang[4], Qingming Huang[1,2,5], and Qi Tian[6]

[1] University of Chinese Academy of Sciences, Beijing, China
[2] Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS, Beijing, China
[3] Kingsoft Cloud, Beijing, China [4] Harbin Inst. of Tech, Weihai, China
[5] Peng Cheng Laboratory, Shenzhen, China
[6] Shenzhen University, Shenzhen, China
hanxinzhe17@mails.ucas.ac.cn, wangshuhui@ict.ac.cn, suchi@kingsoft.com, wgzhang@hit.edu.cn, qmhuang@ucas.ac.cn, wywqtian@gmail.com

Abstract. Visual reasoning is crucial for visual question answering (VQA). However, without labelled programs, implicit reasoning under natural supervision is still quite challenging and previous models are hard to interpret. In this paper, we rethink implicit reasoning process in VQA, and propose a new formulation which maximizes the log-likelihood of joint distribution for the observed question and predicted answer. Accordingly, we derive a Temporal Reasoning Network (TRN) framework which models the implicit reasoning process as sequential planning in latent space. Our model is interpretable on both model design in probabilist and reasoning process via visualization. We experimentally demonstrate that TRN can support implicit reasoning across various datasets. The experimental results of our model are competitive to existing implicit reasoning models and surpass baseline by a large margin on complicated reasoning tasks without extra computation cost in forward stage.

Keywords: Visual Question Answering · Implicit Reasoning · Temporal Reasoning Network · Explanable Machine Learning

1 Introduction

Recent advances in deep learning allow us to investigate emerging research themes lying at the intersection between vision and language. Visual Question Answering (VQA) [3] is a representative task that aims to get an open-ended answer given an image and a natural language question. Since VQA requires high-level understanding of images and the associated questions, visual reasoning is required to provide primitives for deriving a good answer and make VQA model more interpretable for better human understanding [2,26,48,27,58,31,23,36,55].

Existing study towards visual reasoning can be divided into two groups. One group of work are conducted on synthetic datasets, *e.g.*, CLEVR [26], making use of external knowledge. Explicit "functional programs" are adopted by modular networks [2,21,20] and neural symbolism [58,49,53] to generated questions

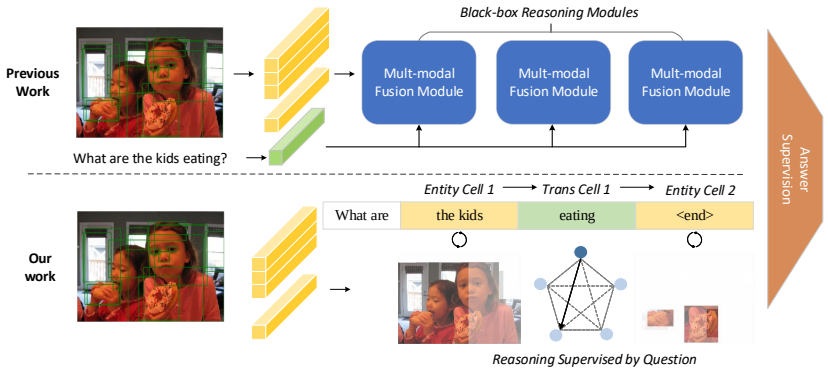


Fig. 1. Comparison between our work and previous work. We regard the question as a hint for reasoning programs, and formulate a Bayesian probabilistic framework for visual reasoning under natural supervision. The highlights in images are sampled from inferred Concrete distribution latent states, which can indicate the most critical image part that should be attended to at each time step

and nearly perfect answers are achieved. These explicit reasoning methods are quite interpretable, but they largely rely on strong assumptions like having labelled programs on questions or labelled entity relations. In another research direction, to solve real-world problem, implicit reasoning via stacked attention modules [48,45,8,15] or graph reasoning [43,22,52,49,36] establish specially designed "black box" neural architectures. Changing in attention maps is thought to be implicit reasoning evidence, and is somewhat a kind of "side effect" of answer classifier optimization. Taking a comparison between implicit and explicit reasoning, it can be found that existing real-world reasoning methods only maximize the likelihood of predicted answers without an understandable reasoning procedure. The multi-hoop attention, obtained by back-propagation, identify components that support a predicted answer, still lacks the ability to explain the reasoning process that achieves a specific answer.

It is known that Bayesian models fall below the ceiling of interpretable machine learning, since they convey a clear representation of the relationships between features and targets. For example, Deep Kalman Filters (DKFs) [38] and Deep Variational Bayes Filters (DVBFs) [30] endow deep models with interpretability inherent to probabilistic graphical models, to make them much more explainable as "Bayesian hybrid transparent" [4]. To unveil the black box of implicit reasoning, we view VQA task from a probabilistic perspective and reformulate a new general Bayesian interpretation for visual reasoning under real-world setting. We pay primary attention to the basic visual reasoning problem formulation and expect to enhance the interpretability of implicit reasoning methods.

Specifically, we reinvestigate the question generation process on synthetic datasets [26,24], which infers that questions clearly convey reasoning programs

and vice versa. Considering that no program labeling is available on real-world data, we assume that there is a set of discrete latent states lying behind input question words, and use them to sequentially describe which part of the image should be attended to at each time step. These latent states act similarly as labelled programs in explicit reasoning models. Based on the latent states, the questions can be regarded as implicit supervision for underlying reasoning programs, as shown in Fig. 1. Instead of only maximizing answer likelihood in previous works, we reformulate an alternative probabilistic interpretation formulation, which maximizes the log-likelihood of joint distribution for the observed question and predicted answer. In this way, the answer predictor and latent states indicating reasoning evidence can be directly optimized simultaneously.

By decomposing the probabilistic formulation, we show that an interpretable reasoning process should have three basic modules, *i.e.*, *State Transition, State Inference* and *Generative Reconstruction*. We also show that recent developments in implicit reasoning can be steadily explained by our probabilistic framework, from one-step State Inference (one-stage fusion [1,16,13,32]) to multi-step State Inference (stacked attention [39,31]) and then to multi-step State Inference with dependent transitions (relation-based methods [14,52,54,22,36]). As a practitioner of the probabilistic framework, we derive a latent sequence model parametrized by a VAE-based neural network, named Temporal Reasoning Network (TRN). We integrate TRN module into existing representative models, such as the one-stage VQA model UpDn [1] and stack-attention model BAN [31]. The injection of TRN on existing models can be regarded as a regularization term in the training stage, and it can be removed in the testing stage without extra computational cost. The results demonstrate that compared to the baseline models, the enhanced models with TRN achieve improved performance and better interpretability without using extra fusion strategies.

It is worth noting that both architecture and loss function in our work are naturally derived from the basic probabilistic formulation and corresponding graphical model. Every term in TRN is conceptually and mathematically interpretable, which further guarantees the interpretability of the whole model. With *Generative Reconstruction* module and latent state sampling, we can also visualize the reasoning process along with question words, which demonstrates that answer prediction procedure of our model is also interpretable. Major contributions are three folds:

- We formulate a new probabilistic interpretation for visual reasoning in the real-world VQA task under natural supervision.
- Following the new probabilistic framework, we propose a sequential latent state model TRN, which is interpretable on both model design and answer prediction.
- TRN can well collaborate with existing models. It can help shallow models like UpDn achieves comparable result compared to state-of-the-art implicit reasoning methods on VQA v2, CLEVR, and CLEVR-Human datasets, and enhance the explanation to existing black-box reasoning models like BAN. Code is available at https://github.com/GeraldHan/TRN.

2 Related Work

2.1 Visual Question Answering and Reasoning

The task of visual question answering is to infer the answer based on the input question and image. Primary methods for VQA mainly focus on better attention mechanisms [56,57,39,14,31] and multi-modal fusion strategies [16,13,32,6,59]. Recent research efforts towards VQA have changed from multi-modal matching to visual reasoning. Existing methods of visual reasoning can be categorized into explicit reasoning and implicit reasoning.

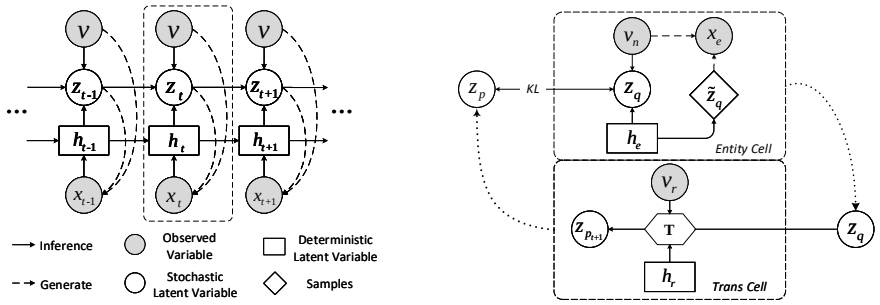
Explicit reasoning are mainly conducted on CLEVR [26] with compositional reasoning procedure. Andreas *et al.* [2] first propose Neural Modular Networks (NMN), which explicitly decompose the reasoning procedure into a sequence of sub-tasks handled by specialized modules. Consequent studies improve this work by proposing better layout policy [21,20], or designing more specific modules [42,27]. Similar to our work, Vedantam *et al.*[53] propose Probabilistic NMN, which provides a probabilistic formulation of NMN and requires a smaller number of teaching examples for layouts. Different from [53], our work focuses on more natural supervision. A more explicit symbolic reasoning method over the object-level structural scene representation is introduced in [58,49,41], which divides the perception and reasoning into two specific stages. Nevertheless, "expert layouts" are needed to supervise the layout policy to get compositional behaviour and good accuracy, which limits their performance on real-world datasets and human asked questions.

Implicit reasoning is extensively studied on real-world datasets like VQA [3]. Bottom-up [1] with object and attribute features extracted by Faster R-CNN [46] is a common baseline. A widely-used approach is to perform reasoning by sequential interactions between image representations and question embeddings [57,23,31,14,15]. Another research line focus on relation reasoning, which can be conducted on a fully-connected graph of objects [48,8]. To better model the interactions between multiple objects, labelled relation or question-conditioned graph representations for images are adopted, then a GCN [34] is used to implicitly infer the interactive representation of objects [43,52,54,22,36]. Implicit reasoning is suitable for real-world setting but much less interpretable.

We build a bridge between implicit and explicit reasoning by introducing latent states behind question words. Thus, the implicit reasoning can be performed with comparable interpretability to explicit reasoning under natural supervision.

2.2 Hybrid Transparent with Bayesian interpretation

The ideas behind variational auto-encoders (VAEs) [33,47] have enabled complex latent dynamical systems like SVAE [29], non-linear SSMs [11,28,12,9] or parametrized deep Markov models [35,51]. These methods combine Bayesian probabilistic graphical models in the embedding space with neural networks to enhance the interpretability of deep models. More similar to our work, model-based reinforcement learning [5,10,7,18] planning in latent space typically assumes access to the low-dimensional states of the environment and plan the



(a) Overall Graphical Model (b) Implimented Model Cell

Fig. 2. Temporal Reasoning Model. (a) is the overall stochastic graphical model for reasoning process, which constructs a latent state z_t underlying each question item x_t conditioned on the given image $\mathbf{v}$ (image is constructed as a graph in practice), simulating labelled programs in explicit models. (b) is the implementation detail for a single reasoning step (the dash block in (a)). Entity Cell models the intra-block reasoning, which infers $q(z_t|h_t, \mathbf{v})$ and $p(x_t|z_t, \mathbf{v})$ based on node features $\mathbf{v}_e$ and entity phrase embedding h_e. Trans Cell infers the prior $p(z_t|z_{t-1}, \mathbf{v})$ for the next time block with a transition function $\mathbf{T}$

dynamics directly in continuous state space, which can be more efficient compared to Bellman backups of traditional reinforcement learning. Recent works TD-VAE [17] and PlaNet [19] further extend to time-sequential planning to solve more complex problems with sequential state observations.

To look inside the implicit reasoning procedure, we formulate a latent sequence model generating implicit policies along with sequential question word inputs. In this way, we construct a deep model with Bayesian hybrid transparent [4] and explicitly model the reasoning procedure without extra-label efforts.

3 Method

3.1 Model Definition

Let $\mathbf{v}$ be image representations and $\mathbf{x}$ is a question comprised of a sequence of L words. The goal of VQA is to predict the best answer $\hat{a} \in \mathcal{A}$, where $\mathcal{A}$ is the answer set. As common practice in the VQA literature, answer prediction can be defined as a classification problem:

$$\hat{a} = \arg\max_{a \in \mathcal{A}} p_\gamma(a|\mathbf{v}, \mathbf{x}) \tag{1}$$

where p_γ denotes the trained classification model.

However, this kind of optimization does not explicitly model the reasoning process. In order to investigate the reasoning process behind the classifier, we

assume there is a sequence of state variable $\mathbf{z} = \{z_t\}_{i=1}^{T}$ that indicates reasoning procedure underlying question $\mathbf{x}$. This time-dependent latent state z_t is decided by the current question item and conditioned on image representations. Fig 2(a) gives a detailed illustration of the whole graphical model for the above reasoning process. According to the graphical model, we treat image representation $\mathbf{v}$ as a global condition, question words $\mathbf{x}$ as sequential observations and assume a general form of fully Bayesian state space model. Our probabilistic formulation specifies the joint distribution $p(a, \mathbf{x}, \mathbf{z}|\mathbf{v})$, and our goal is to find an approximation for model evidence $p(a, \mathbf{x}|\mathbf{v})$ with respect to the posterior distribution $p(\mathbf{z}|\mathbf{x}, \mathbf{v})$. The log marginal evidence probability can be decomposed as

$$\log p(a, \mathbf{x}|\mathbf{v}) = \log p(\mathbf{x}|\mathbf{v}) + \log p(a|\mathbf{x}, \mathbf{v}) \tag{2}$$

Thus, the above model can be divided into two separate parts. The latter part is the answer classifier, similar to traditional VQA models. The former part models the **Temporal Reasoning** process optimized by maximum data likelihood of observed question $\mathbf{x}$, where the underlying latent states $\mathbf{z}$ can be optimized as latent variables via variational inference. Explicitly modelling reasoning process with such a fully probabilistic formulation and injecting it into existing methods are the main contributions of this work. The learning diagram will be detailed in Section 3.2.

3.2 Learning

Temporal Reasoning is to optimize log-likelihood of the question $\log p(\mathbf{x}|\mathbf{v})$. Following Bayesian rule, it can be auto-regressively decomposed as $\log p(\mathbf{x}|\mathbf{v}) = \sum_t \log p(x_t|x_{1:t-1}, \mathbf{v})$. For a given time step t, we decompose the log marginal probability with respect to the variational posterior $q(z_t, z_{t-1}|x_t, \mathbf{v}, x_{1:t-1})$ as

$$\log p(x_t|x_{1:t-1}, \mathbf{v}) = KL(q(z_t, z_{t-1})\|p(z_t, z_{t-1})) + \mathcal{L}_t \tag{3}$$

where $q(z_t, z_{t-1})$ is the short form of $q(z_t, z_{t-1}|\mathbf{v}, x_{1:t})$ which is a inference posterior distribution for latent states z_t, while $p(z_t, z_{t-1})$ is the abbreviation for $p(z_t, z_{t-1}|\mathbf{v}, x_{1:t-1})$ which is a corresponding generative prior distribution.[1] $\mathcal{L}_t$ is the evidence lower bound (ELBO) of data likelihood at time step t, which is

$$\begin{aligned} \mathcal{L}_t = \mathbb{E}_{(z_{t-1}, z_t)\sim q(z_t, z_{t-1})} [\log p(x_t|z_t, z_{t-1}, x_{1:t-1}, \mathbf{v}) \\ + \log p(z_t, z_{t-1}|\mathbf{v}, x_{1:t-1}) - \log q(z_t, z_{t-1}|\mathbf{v}, x_{1:t})] \end{aligned} \tag{4}$$

Considering the Markov assumption underlying the graphical model in Fig. 2(a), we can simplify $p(x_t|z_t, z_{t-1}, x_{1:t-1}, \mathbf{v}) = p(x_t|z_t, \mathbf{v})$. Moreover, following the Bayes rule, we can decompose $q(z_t, z_{t-1}|\mathbf{v}, x_{1:t})$ and $p(z_t, z_{t-1}|\mathbf{v}, x_{1:t-1})$ as

$$\begin{aligned} p(z_t, z_{t-1}|\mathbf{v}, x_{1:t-1}) &= p(z_{t-1}|x_{1:t-1}, \mathbf{v}) p(z_t|z_{t-1}, \mathbf{v}) \\ q(z_t, z_{t-1}|\mathbf{v}, x_{1:t}) &= q(z_t|x_{1:t}, \mathbf{v}) q(z_{t-1}|z_t, x_{1:t}, \mathbf{v}) \end{aligned} \tag{5}$$

[1] Following equations will use the same shorten expressions for convenience. Both distributions are parametrized as neural networks in our work. p indicates generate distributions, while q refers to inference distributions.

Similar to [19], to simplify the model, joint representation $p(x_{1:t})$ is deterministically encoded with a Recurrent Neural Networks (RNNs) as

$$h_t = \mathrm{RNN}(x_t, h_{t-1}) \tag{6}$$

where $h_t \in \mathbb{R}^{d_q}$ is a deterministic variable encoding all history observations. Therefore, Eq. 4 can be decomposed as:

$$\begin{aligned}\mathcal{L}_t = \mathbb{E}_{\substack{z_t \sim q(z_t) \\ z_{t-1} \sim q(z_{t-1})}} \Big[& \log p(x_t|z_t, \mathbf{v}) + \log p(z_{t-1}|h_{t-1}, \mathbf{v}) + \log p(z_t|z_{t-1}, \mathbf{v}) \\ & - \log q(z_t|h_t, \mathbf{v}) - \log q(z_{t-1}|z_t, h_t, \mathbf{v}) \Big]\end{aligned} \tag{7}$$

where $q(z_t)$ and $q(z_{t-1})$ abbreviate $q(z_t|h_t, \mathbf{v})$ and $q(z_{t-1}|z_t, h_t, \mathbf{v})$ respectively.

Answer Classification is similar to existing methods. After Temporal Reasoning, the final state b_T is fed into a Multi-layer Perception (MLP) to predict the answer. Thus, the likelihood of answer can be approximated by a deterministic classifier:

$$\log p(a|\mathbf{x}, \mathbf{v}) \triangleq \mathcal{U} = f(a|b_T) \tag{8}$$

The training objective is to maximize the lower bound of joint data likelihood:

$$\arg\max_{\theta,\phi,\nu} \left[\mathcal{U}(f;\nu) + \sum_{t=1}^{T} \mathcal{L}_t(p, q; \theta, \phi) \right] \tag{9}$$

where the Answer Classifier is parametrized as f_ν, generative distribution and inference distribution in Temporal Reasoning are parametrized as p_θ and q_ϕ respectively.

As shown above, all terms in the loss function are completely derived by variational inference applied to Eq. 2 under a basic latent state assumption. Therefore, all parts of our framework can be mathematically explained from probabilistic perspective, which achieves our interpretability on model design.

3.3 Intuitive Explanation

Eq. 7 derived from the graphical model is parametrized as four different modules. This section will provide a more intuitive explanation behind mathematical derivation. It can further reveal the interpretability of our method.

Generative Reconstruction $\log p(x_t|z_t, \mathbf{v})$ indicates that currently observed word x_t can be reconstructed from corresponding latent state z_t and global condition $\mathbf{v}$. We measure it by Binary Cross Entropy (BCE) between input x_t and reconstructed $\tilde{x}_t$. This term performs external supervision via the input questions.

State Transition $\log p(z_t|z_{t-1}, \mathbf{v})$ predicts a prior distribution for z_t based on former state z_{t-1}. It can be regarded as forward transition in latent space under Markovian assumption. It can guarantee the time-dependency in reasoning process.

State Inference $\log q(z_t|h_t, \mathbf{v})$ indicates that the posterior of latent state distribution $q(z_t)$ depends on history observations h_t and visual features $\mathbf{v}$. Since $p(z_{t-1}|h_{t-1}, \mathbf{v})$ has a consistent dependency with the posterior $q(z_{t-1}|h_{t-1}, \mathbf{v})$ at former time step $t-1$, we approximate generative distribution $p(z_{t-1}|h_{t-1}, \mathbf{v}) = q_\phi(z_{t-1}|h_{t-1}, \mathbf{v})$ without loss of information.
Backward Transition $\log q(z_{t-1}|z_t, h_t, \mathbf{v})$ indicates the former state z_{t-1} can be re-inferenced from the current state and observations. This term has a similar facility with State Inference but is hard to model, so we ignore this term in practice to simplify our model.

Comparing with recent proposed VQA methods, this probabilistic formulation can help explain recent developments in implicit reasoning. The attention mechanism can be viewed as a type of *State Inference* in latent space. From one-stage attention/fusion model [16,1] to stacked attention methods [39,31], the performance is largely improved due to the introduction of *multi-step State Inference*. Recent proposed relation-based methods [14,52,54,22,36] further strengthen the dependences between stacked modules, that can collaborate with the function of *State Transition* in our formulation. Moreover, stronger fusion strategies can help establish a more informative latent space. From this perspective, implicit reasoning is indeed sequentially supervised by both question and answer but has not been explicitly modelled before.

3.4 Parametrization and Implementation

Following the instruction of probabilistic formulation, we implement the temporal reasoning process as a VAE based latent sequence model, named Temporal Reasoning Network (TRN). It should be stressed that TRN is not a fixed network. We implement it as complementary modules upon existing baseline models based on the proposed graphical model.

Latent State distribution. In order to reveal the reasoning procedure underlying question words and fairly compare with attention-based baselines, we assume latent states following Concrete distribution [40]

$$q(z_t) = \mathcal{C}(\pi_t, \tau) \tag{10}$$

where $\pi_t \in \mathbb{R}^K$ indicates the decision evidence at time step t and τ is the superparameter for temperature. In practice, we use Exponential Concrete distribution for more stable calculation of logarithm probability. We sample $\tilde{z}_t$ from $q(z_t)$ using Gumbel Softmax trick [25] for gradient back-propagation. The distribution function and calculation of log-probability will be provided in supplementary material.

Feature Parametrization. To better implement the reasoning process, we reformulate the image representation as a graph, where $\mathbf{v}_n \in \mathbb{R}^{K \times d_n}$ are node features indicating K objects, and $\mathbf{v}_r \in \mathbb{R}^{K \times K \times d_r}$ denote edge features of relations between nodes. Moreover, regarding every question word as a state is time-consuming and hard to ground in $\mathbf{v}$. We extract noun phrases from the input question by open-sourced spaCy as *entity phrases*, while the phrase between

noun chunks as *transition phrases*. Parsing question by phrases can obtain more semantic information and save computational cost. Following [37], this phrase representation is encoded by Bi-directional Gated Recurrent Unit (BiGRU) to capture the context information.

$$\begin{aligned}(\overrightarrow{w_{1:L}}; \overleftarrow{w_{1:L}}) &= \text{Bi-GRU}(\mathbf{x}) \\ h_e^t = [\overrightarrow{w_{e_t}}; \overleftarrow{w_{s_t}}],\ h_r^t &= [\overrightarrow{w_{s_{t+1}}}; \overleftarrow{w_{e_t}}],\ h_e^T = [\overrightarrow{w_L}; \overleftarrow{w_0}]\end{aligned} \tag{11}$$

where e_t and s_t are start and end location of the t-th entity phrase, $h_e^t \in \mathbb{R}^{d_q}$ and $h_r^t \in \mathbb{R}^{d_q}$ denote entity embeddings and transition embeddings at time step t. We treat the global GRU output $[\overrightarrow{w_L}; \overleftarrow{w_0}]$ as the last entity embedding h_e^T. Since the number of entity phrases are much smaller than that of the question words, parsing by phrase not only largely save computational cost but obtain more semantic information that can be grounded in the image as well.

Implementation Details. We implement TRN as an injected module to both classical one stage method Bottom-up Top-down Attention (UpDn) [1] and widely used implicit reasoning method Bilinear Attention Network (BAN) [31]. A single Temporal Reasoning cell is divided into two cells, *i.e.*, *Entity Cell* and *Trans Cell*. As shown in Fig. 2(b), Entity Cell models the State Inference $q(z_t|h_t, \mathbf{v})$ and Generative Reconstruction term $\log p(x_t|z_t, \mathbf{v})$. Trans Cell infers State Transition term $p(z_t|z_{t-1}, \mathbf{v})$ for the next time block, which is modelled as one-step Markovian transition on the graph. More implement details can be found in Section 3.4 and Algorithm 1 in supplementary material.

It can be seen that the only connection between each reasoning block is just the K-L Divergence of $p(z_t|z_{t-1}, \mathbf{v})$ and $q(z_t|h_t, \mathbf{v})$. For a fair comparison with baseline, no more extra fusion strategies are used in TRN. Therefore, the whole TRN can be regarded as a regularization term in the training stage and can be removed in test stage, which would not bring extra computational cost compared to original methods.

4 Experiments

4.1 Datasets

VQA 2.0 is a commonly used VQA dataset composed of real-world images from MSCOCO with the same train/validation/test splits. Following previous works, we take the answers that appeared more than 9 times in the training set as candidate answers, which produces 3129 answer candidates.

CLEVR is a synthetic dataset, consisting of visual scenes with simple geometric shapes with complicated relational questions like "*What size is the cylinder that is left of the brown metal thing that is left of the big sphere? "*. It is the most commonly used dataset for visual reasoning that requires the model's long-chain reasoning ability. Since this work focuses on VQA under natural supervision, we only use question-answer pairs annotation in CLEVR to evaluate our reasoning ability in complicated questions.

CLEVR Human dataset consists of human-generated questions for CLEVR images, which can test the model generalization for real-world questions, since all the questions are generated from programs in the original dataset.

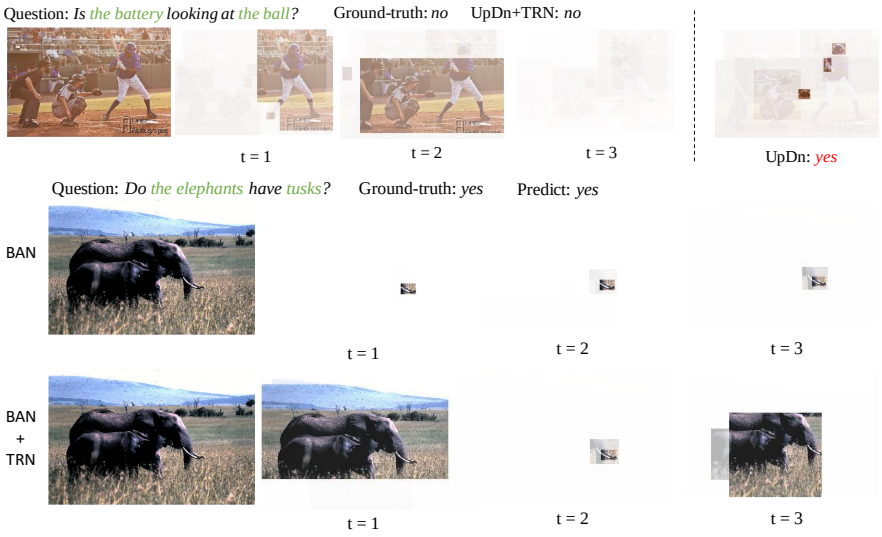


Fig. 3. Examples from VQA v2.0 val, which visualize the change of latent states across TRN Blocks. The highlighted regions are the most important areas that tend to be sampled. The upper example is from UpDn+TRN, the model first finds the batter and then searches the area that he is looking at (no balls). In practice, our model can directly output $t = 3$ and the answer *no*. UpDn gives a wrong answer and attends at the bat. The lower example is from BAN+TRN. The model first finds the elephants and tusks, then finds elephants with tusks. Without dependence between stacked modules, the attention maps of BAN almost remain unchanged. Overlooking the elephants and directly locating tusks may lead to over-fitting on dataset bias and be prone to fail in queries like "tigers with tusks". More examples can be found in the Supplementary

4.2 Evaluation on Real-world Datasets

Experiment Settings. We use the object proposal feature provided by [1]. The node features $\mathbf{v}_n \in \mathbb{R}^{36\times2048}$ consists of 36 objects features, each object feature v_i is a local visual feature vector $o_i \in \mathbb{R}^{2048}$ extracted by Faster R-CNN [46]. The edge features $\mathbf{v}_r \in \mathbb{R}^{36\times36\times1024}$ are concatenation of corresponding node feature transforming to 1024-dim. For question $\mathbf{x}$, each word x_t is first initialized by 300-dim GloVe word embeddings [44], then fed into a Bi-GRU. The final representation h_e^t and h_r^t are 1024-dim phrase embeddings with context information. The number of Temporal Reasoning Blocks is set as 3.

Comparison with Baseline Models. Since real-world dataset VQA v2.0 does not contain too many questions that need reasoning, our TRN only sightly improves the performance compared to baselines, as shown in Table 1. The primary function of TRN in VQA v2.0 is to obtain an interpretable reasoning process.

As shown in Fig. 3, one-stage models like UpDn does not perform well on questions considering multiple objects, our model can deal with this drawback and improve the performance on questions that require reasoningDespite inte-

Table 1. Model accuracy on the VQA v2.0 benchmark (open-ended setting on the test-dev and test-std split). Methods with * are reimplemented by ourselves. Red colour highlights the best performance, and blue numbers are the second place

Method	test-dev				test-std
	Y/N	Num.	Other	All	
MUTAN [6]	82.88	44.54	56.50	66.01	66.38
MuRel [8]	84.77	49.84	57.85	68.03	68.41
Dyna Tree [52]	84.28	47.78	59.11	68.19	68.49
DFAF [14]	86.09	53.53	60.49	70.22	70.34
QCG [43]	82.91	47.13	56.22	56.45	66.17
RAMEN [50]	-	-	-	-	65.96
UpDn* [1]	82.64	45.51	57.21	65.82	65.91
BAN-3* [31]	84.68	50.71	58.56	68.43	68.47
UpDn + TRN	83.83	45.61	57.44	67.00	67.21
BAN-3 + TRN	84.59	50.23	58.64	68.38	68.76

grate multi-step attention, the stacked attentions in BAN are almost independent. Although attending to the right objects, it cannot provide explainable evidence for reasoning. With the help of TRN, the visualization of sampling in latent spaces from BAN+TRN are closer to human understanding.

Comparison with Other Methods. We compare with two recently proposed reasoning models with multi-step attention and fusion strategy (MUTAN [6], MuRel [8]), three models focus on relation reasoning (QCG [43], Dyna Tree [52], DFAF [14]) and RAMEN [50] that claims to work on both real-world and synthetic datasets. All methods are trained with both train and validation split without model ensemble.Our method achieves comparable performance and BAN-3+TRN is the second place in VQA v2.0 test-std split.

Among these methods, DFAF uses at most 100 region proposals and achieves the best single model performance[2]. MuRel is a variant of BAN, and the reasoning process can also be visualized. The major module DynamicIntraMAF in DFAF and Pairwise module proposed in MuRel play similarily as the State Transition term in TRN. Our probabilistic formulation can help enhance their interpretability with Bayesian transparency.

4.3 Evaluation on Synthetic Datasets

Experiment Settings. For CLEVR dataset, The node representations $\mathbf{v}_n \in \mathbb{R}^{15\times 18}$ include at most 15 proposal features, each object feature v_i is an 18-dim output of object attribute extractor provided in NS-VQA [58], which refers to the shape, colour, material, and 3-dim coordinate position of the proposal object. Since this representation contains enough semantics and the relationships

[2] Our reproduction with 36 proposals only gets 67.69% accuracy on test-std.

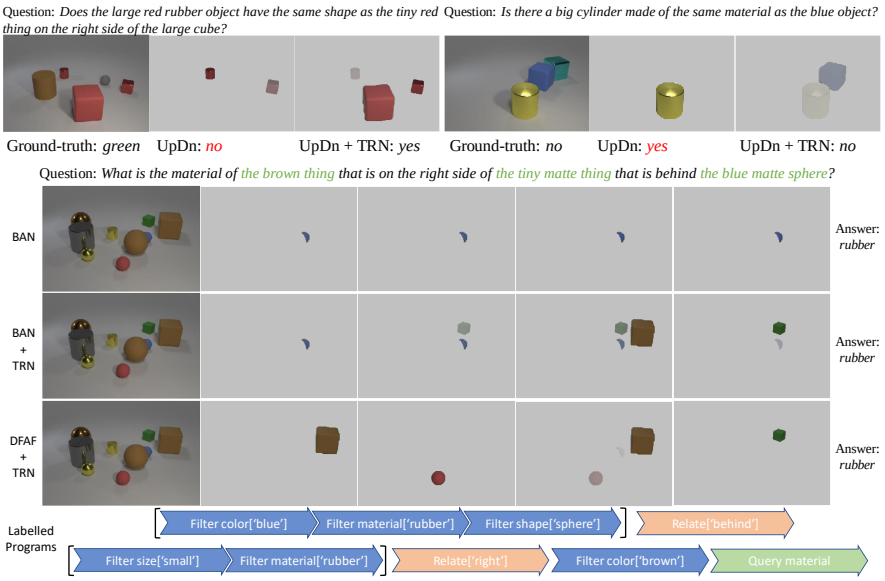


Fig. 4. Examples from our TRN on validation split of CLEVR and CLEVR-Humans. The upper row shows the last latent state between UpDn and UpDn+TRN. The latter two examples are reasoning process visualization of BAN, BAN+TRN, and DFAF+TRN. The depth of colour indicates sample value from latent space. Though outputting the same answers, BAN provides the right answer but wrong evidence, while our model can catch up with the right piece of evidences that is much closer to labelled programs. DFAF+TRN can better locate informative objects, but the time dependence in reasoning process is relatively worse due to strong fusion strategy

between objects are simple, the edge feature $v_r^{i,j} \in \mathbb{R}^{18}$ are just defined as $v_n^i - v_n^j$. Question $\mathbf{x}$ is randomly initialized, and we get final phrase representations $h_t \in \mathbb{R}^{600}$ after the same operations with real-world setting. The temperature of Concrete distribution is 2.0. The maximum entity number is set to be 5 in experiments. For CLEVR, we train on CLEVR train split and test on validation split. For CLEVR-Humans, we first pre-train our model on CLEVR train split, and then fine-tune on CLEVR-Humans train split.

Comparison with Baseline Models. Most questions in CLEVR deal with multiple objects and require long-time reasoning ability. As shown in Table 2, TRN can surpass one-stage baseline UpDn by a large margin on both CLEVR and CLEVR-Humans. With help of the additional reasoning process in TRN, many failure cases in UpDn can be corrected. The visualizations of the last latent state in Fig. 4 further demonstrate the effectiveness of our method. Moreover, TRN can be removed in the test stage, which means this improvement does not require extra computational cost.

Table 2. Model accuracy comparison on CLEVR and CLEVR-Humans. Methods with * is reproduced by ourselves. Red colour highlights the best performance, and blue numbers are the second place

Method	CLEVR	CLEVR-Humans
Film [45]	97.6%	75.9%
RN [48]	95.5%	57.6%
MAC [23]	98.0%	50.2%
RAMEN [50]	96.9%	57.8%
LCGN [22]	97.9%	-
UpDn*[1]	78.1%	56.6%
BAN-5* [45]	83.1%	60.5%
DFAF-5* [14]	95.5%	63.2%
UpDn + TRN	87.7%	69.5%
BAN-5 + TRN	85.2%	65.4%
DFAF-5 + TRN	96.7%	72.9%

Table 3. Ablation study for UpDn+TRN on CLEVR validation set. SI, ST and GR stand for *State Inference*, *State Transition* and *Generative Reconstruction*, respectively. Components in TRN are indivisible. On CLEVR val, UpDn+TRN degrades to original UpDn without State Inference or State Transition. UpDn+TRN without Generative Reconstruction is similar with BAN, which can improve to 82.3% accuracy but poor visualization

SI	ST	GR	CLEVR val
✓		✓	78.1%
	✓	✓	77.6%
✓	✓		84.3%
✓	✓	✓	87.7%

Visualization of the reasoning process reveals that BAN does not really understand the question. As shown in Fig. 4, although giving correct answers, BAN cannot provide understandable reasoning process[3], and sometimes even attends to wrong evidences. On the contrary, BAN+TRN can offer more explainable reasoning as visualized which is much closer to labelled programs and human understanding. The visualization is much more apparent than on VQA v2.0 that TRN not only grounds noun phrases in image, but illustrates time-dependent reasoning process. Moreover, the final results of BAN+TRN are worse than UpDn+TRN. We speculate that the major reason may be the high-level features we use for node representation. Since BAN mainly focuses on multi-modal fusion, this 18-dim vector may be too abstracted for multi-modal fusion.

Comparison with Other Methods. We compare our model with four previous works (Film [45], RN [48], MAC [23], LCGN [22]) that do not use any functional program information. Recent proposed method LCGN does not report their performance on CLEVR-Humans.

Due to long-time fusion strategies, most of these methods are better than our model on CLEVR. To verify this, we conduct TRN on DFAF [14], which adopts inter-intra attention fusion strategy across stacked modules. The implementation details can be found in the Supplementary. As shown in Table 2, DFAF-5+TRN achieves 96.7% accuracy, which is comparable to state-of-the-art implicit reasoning models specially designed for CLEVR. This indicates that stronger fusion strategies can help establish more informative latent space, which would be a crucial complement for complicated reasoning process. However, visualiza-

[3] The object attention is Softmax of the sum of $\mathcal{A}$ along question dimension.

tion of DFAF+TRN shown in Fig. 4 is not as understandable as BAN+TRN, because complex fusion has overly strong fitting ability that can integrate information without considering time dependences. Moreover, we achieve better performance on CLEVR-Humans. Despite not using labelled programs, models with over-parametrized fusion largely rely on the fixed grammar of input questions, resulting in over-fitting.

4.4 Discussion

Ablation study. As shown in Table 3, components in TRN are indivisible and derived from an integrated process with question supervision. Moreover, 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). A more detailed ablation study is provided in Section 4.1 in the Supplementary.
Failure cases. Failure cases are provided in Section 6 in the Supplementary. TRN performs poorly on counting problems. This may be a result of the intrinsic weakness of Concrete distribution, which tends to sample one-hot vector and ignore items that are relatively unimportant. This shortcoming also exists in attention mechanism. It can still be improved by choosing better latent state distributions or specially designed modules. Another shortcoming is in dealing with adverbial problems. Reasoning phrase-by-phrase may fail to catch relationships between entities that are distant in the question. Although stronger fusion strategies can help catch up with enough information, it may break the chain of reasoning process due to overly strong fitting ability. How to model the long-time dependences is a common challenge for latent sequence models. Modelling an interpretable fusion strategy may be the future work that needs to be done.

5 Conclusion

Reasoning in VQA under natural supervision is a very challenging task due to super asymmetric information. In this work, we analyse real-world VQA task from a new perspective, and propose a new probabilistic formulation that can explicitly model the reasoning process without extra program labeling. Experiments on both real-world and synthetic datasets demonstrate our model's effectiveness and interpretability. We hope such a probabilistic formulation can provide guidance on further advancements in problems with insufficient natural supervision or other tasks that need multi-step programming. In future work, we will devote our efforts to learning interpretable models for complicated vision-language tasks by combining knowledges.

Acknowledgement. This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0102003, in part by National Natural Science Foundation of China: 61672497, 61620106009, 61836002, 61931008 and U1636214, and in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013. Authors are grateful to Kingsoft Cloud for support of free GPU cloud computing resource and Yuecong Min for fruitful discussion.

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) 3, 4, 8, 9, 10, 11, 13
2. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Neural module networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 39–48 (2016) 1, 4
3. 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) 1, 4
4. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al.: Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information Fusion **58**, 82–115 (2020) 2, 5
5. Banijamali, E., Shu, R., Ghavamzadeh, M., Bui, H., Ghodsi, A.: Robust locally-linear controllable embedding. arXiv preprint arXiv:1710.05373 (2017) 4
6. Ben-Younes, H., Cadene, R., Cord, M., Thome, N.: Mutan: Multimodal tucker fusion for visual question answering. In: Proceedings of the IEEE international conference on computer vision. pp. 2612–2620 (2017) 4, 11
7. Buesing, L., Weber, T., Racaniere, S., Eslami, S., Rezende, D., Reichert, D.P., Viola, F., Besse, F., Gregor, K., Hassabis, D., et al.: Learning and querying fast generative models for reinforcement learning. arXiv preprint arXiv:1802.03006 (2018) 4
8. Cadene, R., Ben-Younes, H., Cord, M., Thome, N.: Murel: Multimodal relational reasoning for visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1989–1998 (2019) 2, 4, 11
9. Chen, T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. In: Advances in neural information processing systems. pp. 6571–6583 (2018) 4
10. Chua, K., Calandra, R., McAllister, R., Levine, S.: Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In: Advances in Neural Information Processing Systems. pp. 4754–4765 (2018) 4
11. Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A.C., Bengio, Y.: A recurrent latent variable model for sequential data. In: Advances in neural information processing systems. pp. 2980–2988 (2015) 4
12. Doerr, A., Daniel, C., Schiegg, M., Nguyen-Tuong, D., Schaal, S., Toussaint, M., Trimpe, S.: Probabilistic recurrent state-space models. arXiv preprint arXiv:1801.10395 (2018) 4
13. Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T., Rohrbach, M.: Multimodal compact bilinear pooling for visual question answering and visual grounding. arXiv preprint arXiv:1606.01847 (2016) 3, 4
14. Gao, P., Jiang, Z., You, H., Lu, P., Hoi, S.C., Wang, X., Li, H.: Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6639–6648 (2019) 3, 4, 8, 11, 13
15. Gao, P., You, H., Zhang, Z., Wang, X., Li, H.: Multi-modality latent interaction network for visual question answering. arXiv preprint arXiv:1908.04289 (2019) 2, 4

16. Gao, Y., Beijbom, O., Zhang, N., Darrell, T.: Compact bilinear pooling. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 317–326 (2016) 3, 4, 8
17. Gregor, K., Papamakarios, G., Besse, F., Buesing, L., Weber, T.: Temporal difference variational auto-encoder. arXiv preprint arXiv:1806.03107 (2018) 5
18. Ha, D., Schmidhuber, J.: World models. arXiv preprint arXiv:1803.10122 (2018) 4
19. Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., Davidson, J.: Learning latent dynamics for planning from pixels. arXiv preprint arXiv:1811.04551 (2018) 5, 7
20. Hu, R., Andreas, J., Darrell, T., Saenko, K.: Explainable neural computation via a stack neural module networks. In: Proceedings of the European conference on computer vision (ECCV). pp. 53–69 (2018) 1, 4
21. Hu, R., Andreas, J., Rohrbach, M., Darrell, T., Saenko, K.: Learning to reason: End-to-end module networks for visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017) 1, 4
22. Hu, R., Rohrbach, A., Darrell, T., Saenko, K.: Language-conditioned graph networks for relational reasoning. arXiv preprint arXiv:1905.04405 (2019) 2, 3, 4, 8, 13
23. Hudson, D.A., Manning, C.D.: Compositional attention networks for machine reasoning. In: ICLR (2018) 1, 4, 13
24. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6700–6709 (2019) 2
25. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016) 8
26. 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) 1, 2, 4
27. Johnson, J., Hariharan, B., van der Maaten, L., Hoffman, J., Fei-Fei, L., Lawrence Zitnick, C., Girshick, R.: Inferring and executing programs for visual reasoning. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2989–2998 (2017) 1, 4
28. Johnson, M., Duvenaud, D.K., Wiltschko, A., Adams, R.P., Datta, S.R.: Composing graphical models with neural networks for structured representations and fast inference. In: Advances in neural information processing systems. pp. 2946–2954 (2016) 4
29. Johnson, M.J., Duvenaud, D.K., Wiltschko, A., Adams, R.P., Datta, S.R.: Composing graphical models with neural networks for structured representations and fast inference. In: Advances in neural information processing systems. pp. 2946–2954 (2016) 4
30. Karl, M., Soelch, M., Bayer, J., Van der Smagt, P.: Deep variational bayes filters: Unsupervised learning of state space models from raw data. arXiv preprint arXiv:1605.06432 (2016) 2
31. Kim, J.H., Jun, J., Zhang, B.T.: Bilinear attention networks. In: Advances in Neural Information Processing Systems. pp. 1564–1574 (2018) 1, 3, 4, 8, 9, 11
32. Kim, J.H., On, K.W., Lim, W., Kim, J., Ha, J.W., Zhang, B.T.: Hadamard product for low-rank bilinear pooling. arXiv preprint arXiv:1610.04325 (2016) 3, 4
33. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013) 4

34. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016) 4
35. Krishnan, R.G., Shalit, U., Sontag, D.: Structured inference networks for nonlinear state space models. In: Thirty-First AAAI Conference on Artificial Intelligence (2017) 4
36. Li, L., Gan, Z., Cheng, Y., Liu, J.: Relation-aware graph attention network for visual question answering. arXiv preprint arXiv:1903.12314 (2019) 1, 2, 3, 4, 8
37. Liu, J., Hockenmaier, J.: Phrase grounding by soft-label chain conditional random field. arXiv preprint arXiv:1909.00301 (2019) 9
38. Lu, G., Ouyang, W., Xu, D., Zhang, X., Gao, Z., Sun, M.T.: Deep kalman filtering network for video compression artifact reduction. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 568–584 (2018) 2
39. Lu, J., Yang, J., Batra, D., Parikh, D.: Hierarchical question-image co-attention for visual question answering. In: Advances In Neural Information Processing Systems. pp. 289–297 (2016) 3, 4, 8
40. Maddison, C.J., Mnih, A., Teh, Y.W.: The concrete distribution: A continuous relaxation of discrete random variables. arXiv preprint arXiv:1611.00712 (2016) 8
41. Mao, J., Gan, C., Kohli, P., Tenenbaum, J.B., Wu, J.: The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. arXiv preprint arXiv:1904.12584 (2019) 4
42. Mascharka, D., Tran, P., Soklaski, R., Majumdar, A.: Transparency by design: Closing the gap between performance and interpretability in visual reasoning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4942–4950 (2018) 4
43. Norcliffe-Brown, W., Vafeias, S., Parisot, S.: Learning conditioned graph structures for interpretable visual question answering. In: Advances in Neural Information Processing Systems. pp. 8334–8343 (2018) 2, 4, 11
44. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014) 10
45. Perez, E., Strub, F., De Vries, H., Dumoulin, V., Courville, A.: Film: Visual reasoning with a general conditioning layer. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018) 2, 13
46. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015) 4, 10
47. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082 (2014) 4
48. Santoro, A., Raposo, D., Barrett, D.G., Malinowski, M., Pascanu, R., Battaglia, P., Lillicrap, T.: A simple neural network module for relational reasoning. In: Advances in neural information processing systems. pp. 4967–4976 (2017) 1, 2, 4, 13
49. Shi, J., Zhang, H., Li, J.: Explainable and explicit visual reasoning over scene graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8376–8384 (2019) 1, 2, 4
50. Shrestha, R., Kafle, K., Kanan, C.: Answer them all! toward universal visual question answering models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10472–10481 (2019) 11, 13
51. Tan, Z.X., Soh, H., Ong, D.C.: Factorized inference in deep markov models for incomplete multimodal time series. arXiv preprint arXiv:1905.13570 (2019) 4

52. Tang, K., Zhang, H., Wu, B., Luo, W., Liu, W.: Learning to compose dynamic tree structures for visual contexts. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019) 2, 3, 4, 8, 11
53. Vedantam, R., Desai, K., Lee, S., Rohrbach, M., Batra, D., Parikh, D.: Probabilistic neural-symbolic models for interpretable visual question answering. arXiv preprint arXiv:1902.07864 (2019) 1, 4
54. Wang, P., Wu, Q., Cao, J., Shen, C., Gao, L., Hengel, A.v.d.: Neighbourhood watch: Referring expression comprehension via language-guided graph attention networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1960–1968 (2019) 3, 4, 8
55. Wei, K., Yang, M., Wang, H., Deng, C., Liu, X.: Adversarial fine-grained composition learning for unseen attribute-object recognition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3741–3749 (2019) 1
56. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. pp. 2048–2057 (2015) 4
57. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.: Stacked attention networks for image question answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 21–29 (2016) 4
58. Yi, K., Wu, J., Gan, C., Torralba, A., Kohli, P., Tenenbaum, J.: Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In: Advances in Neural Information Processing Systems. pp. 1031–1042 (2018) 1, 4, 11
59. Yu, Z., Yu, J., Xiang, C., Fan, J., Tao, D.: Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering. IEEE transactions on neural networks and learning systems **29**(12), 5947–5959 (2018) 4

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

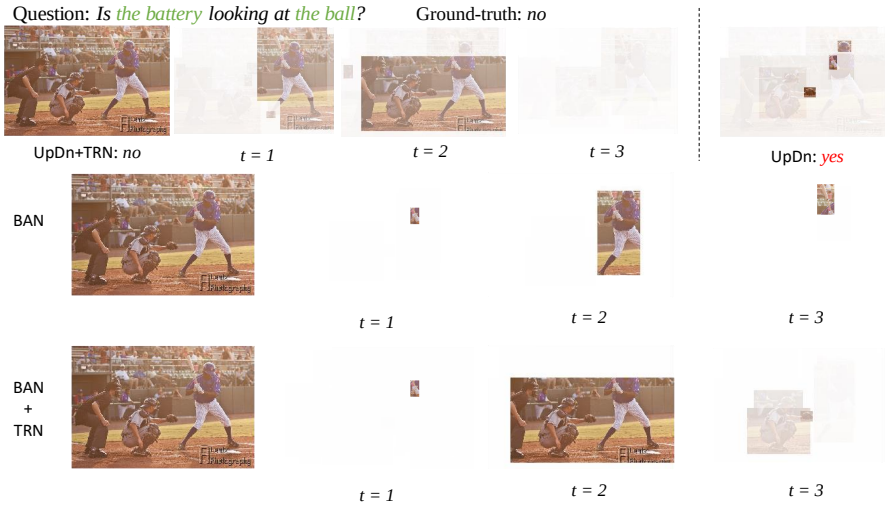
Level of interpretability is too subjective. it is too difficult to quantitatively compare the visual vector with functional program in language. Apart form qualitative comparison with labelled program (Figure 4), we also design a *Subjective Blind Test* for more convincing analysis.

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:

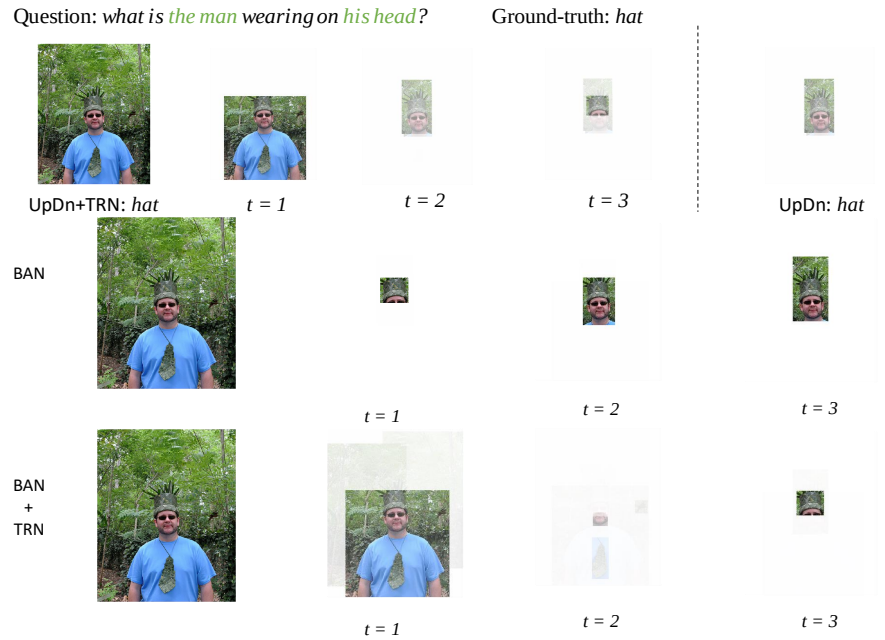
- 0: Totally wrong evidence
- 1: Noisy answer evidence or reasoning evidence, no reasoning process
- 2: Clear answer evidence, no reasoning evidence
- 3: Clear answer evidence, noisy reasoning evidence
- 4: Clear answer evidence and reasoning evidence, somehow understandable reasoning process
- 5: Clear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

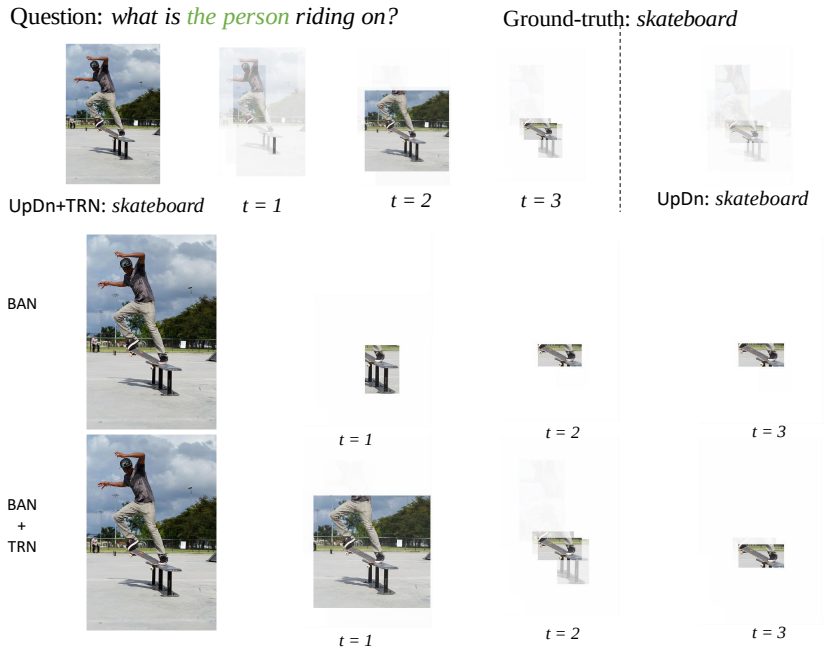


Fig. 1. Extra Examples From VQA v2. The first row is the comparison between UpDn and UpDn+TRN. TRN can display the reasoning process and improve performance without extra computational cost in the testing stage. Examples in the lower rows are the comparison between BAN and BAN+TRN. TRN can boost the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

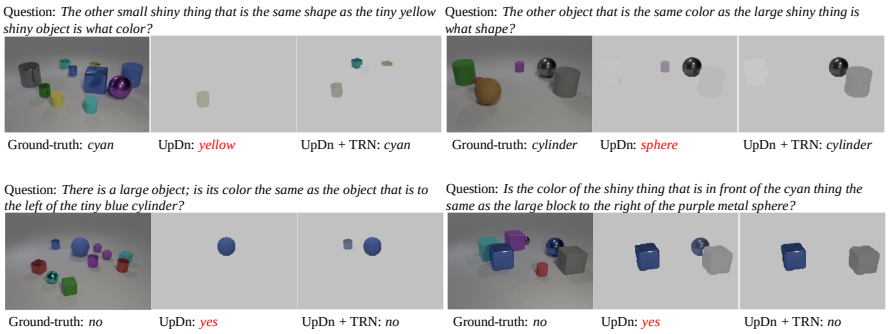


Fig. 2. More comparisons between UpDn and UpDn+TRN. TRN can offer right answers for questions that fail to be answered by UpDn.

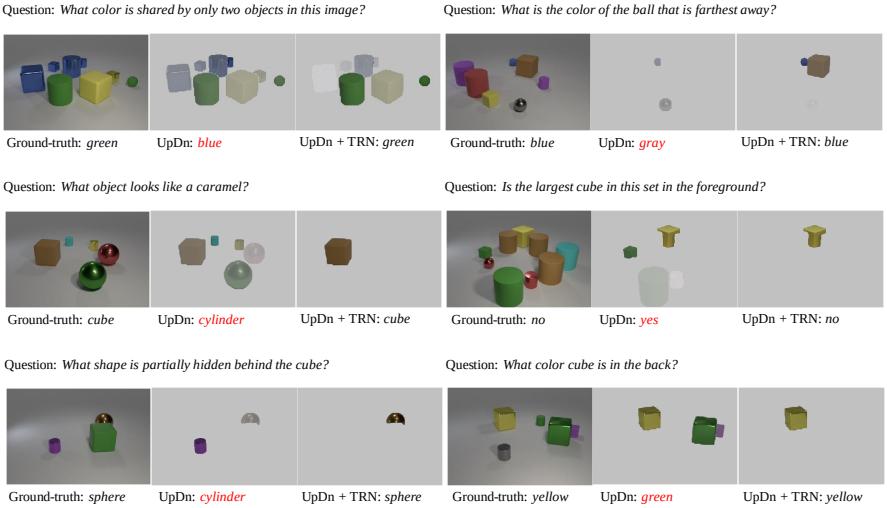
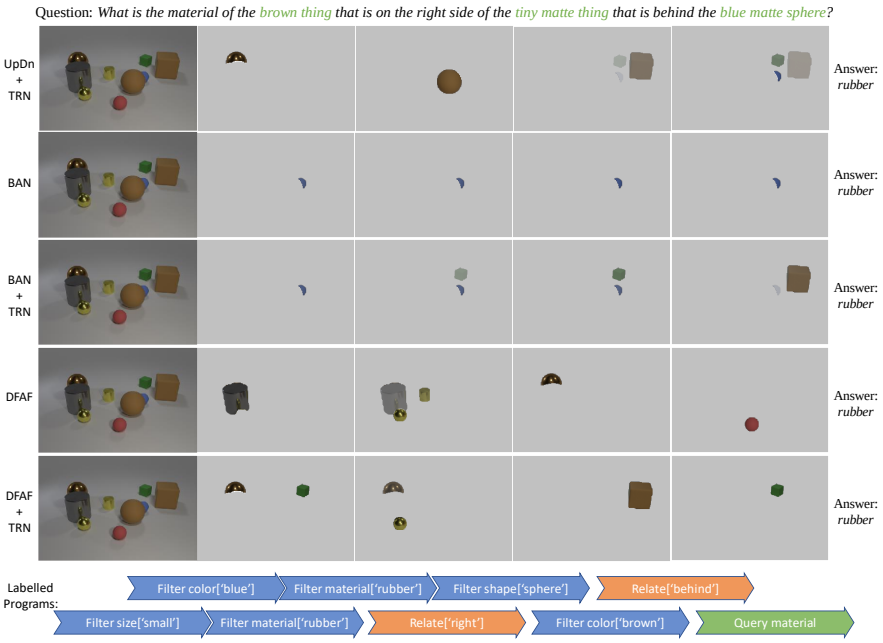
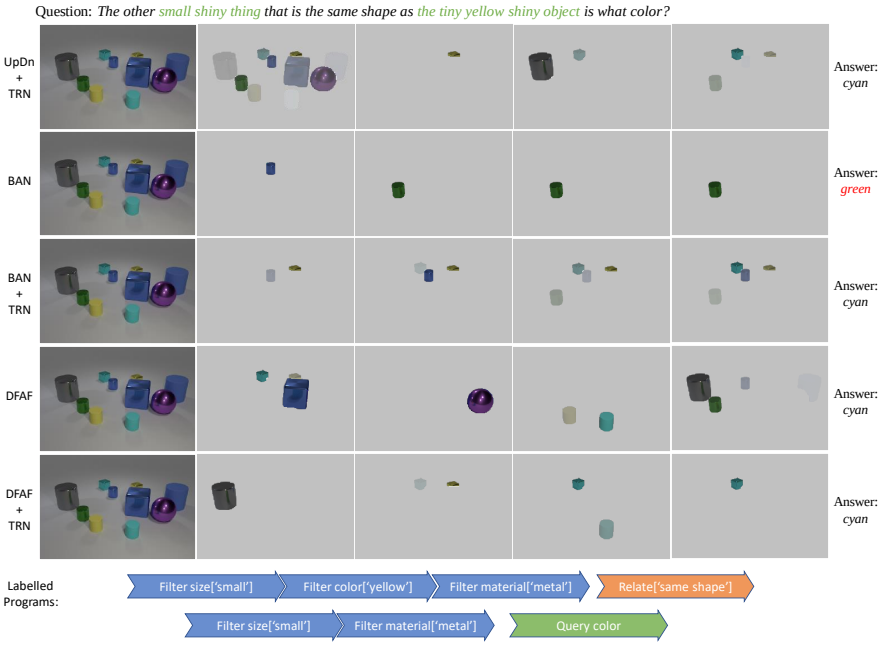


Fig. 3. Comparisons between UpDn and UpDn+TRN on CLEVR-Humans.


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

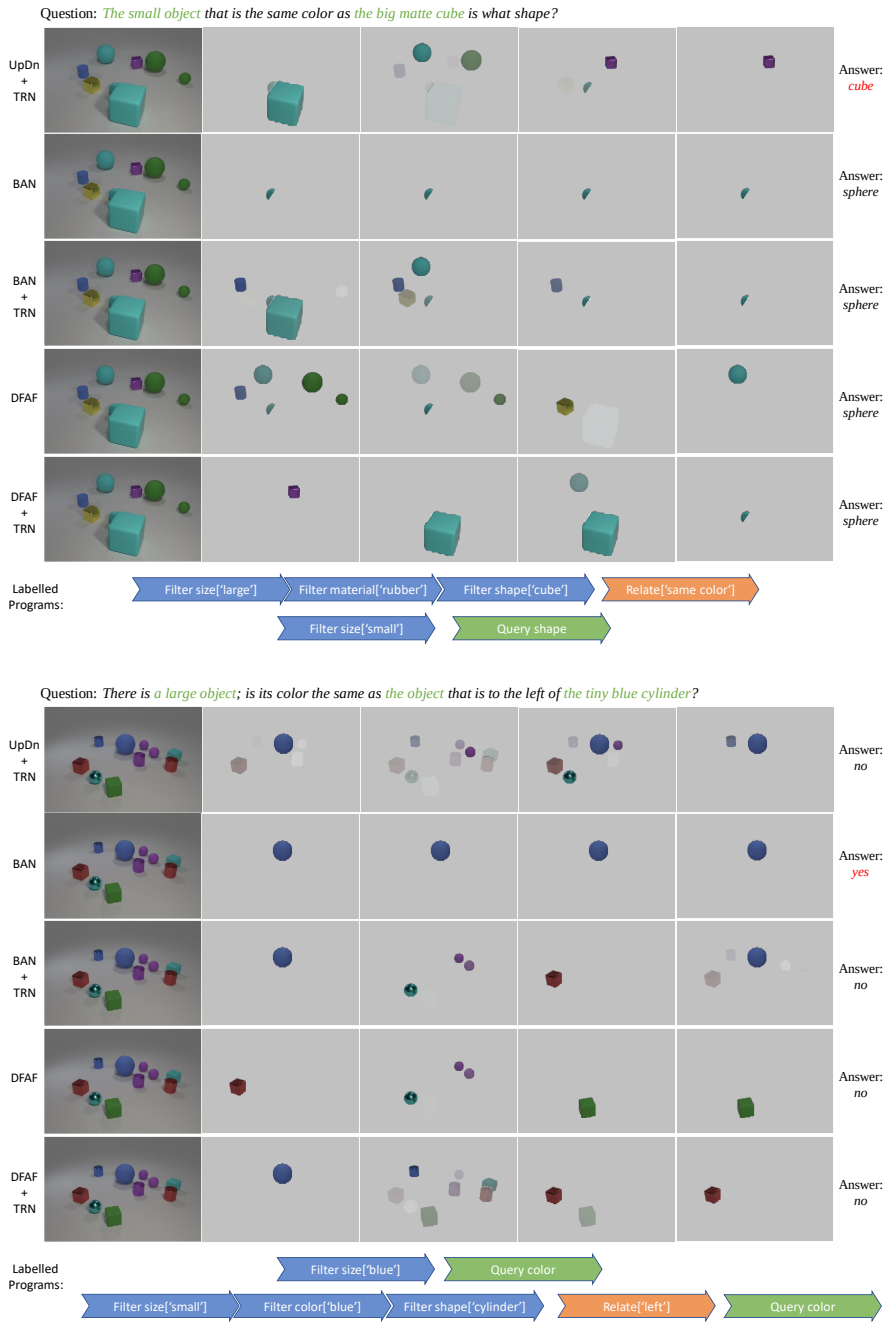


Fig. 4. Examples for visualizing reasoning process on CLEVR. TRN is closer to labelled programs and human understanding.

6 Failed Cases

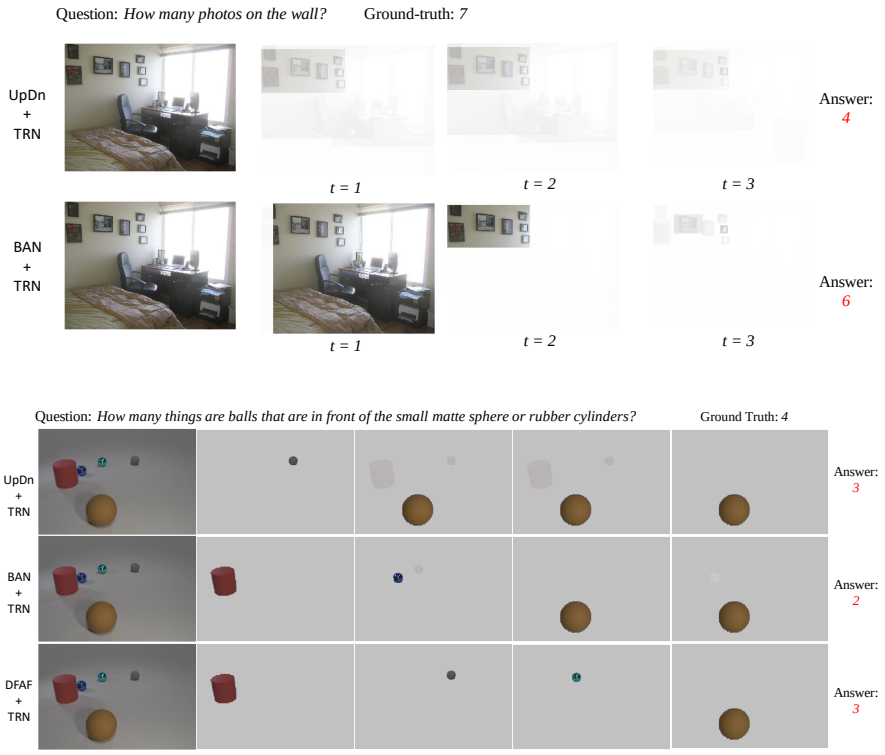


Fig. 5. Failed cases for counting problems. This is a common challenge for attention-based methods

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.

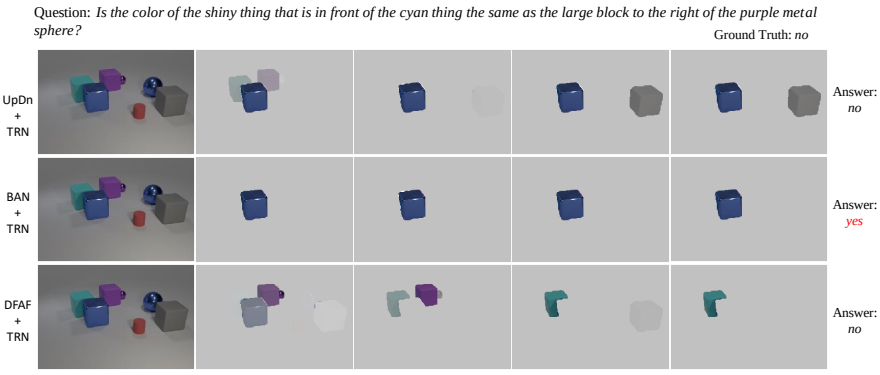


Fig. 7. Failed case for adverbial problem in complex questions. The performance could be improved by fusion strategy like DFAF, but is still poor in the reasoning process visualization.

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