A Confidence-based Multipath Neural-symbolic Approach for Visual Question Answering

Yajie Bao[1] , **Tianwei Xing**[2] and **Xun Chen**[2]
[1]University of Georgia
[2]Samsung Research America
yajie.bao@uga.edu, {t.xing, xun.chen}@samsung.com

Abstract

Neural-symbolic (NS) learning provides an efficient approach to visual question answering (VQA) by combining the advantages of neural network learning and symbolic reasoning. However, the uncertainty of the neural networks (NN) learning in the existing NS methods has not been considered, and one single answer is provided for a question without confidence evaluation. To tackle this problem, we propose a confidence-based NS (CBNS) framework to evaluate the confidence of the NN modules based on uncertainty quantification and make inferences based on the confidence evaluations. Specifically, CBNS includes a probabilistic question parser which generates multiple program candidates with confidence evaluations. CBNS also includes a probabilistic scene perception module which provides object-based scene representation and confidence evaluations for each attribute of objects in an image. The object-based scene representation and the programs with confidence evaluations are used for evaluating the confidence of answers through the inference process. The proposed framework is model-agnostic and compatible with mainstream NS VQA architectures. Experiments on CLEVR demonstrate that the proposed framework enables confidence-based reasoning for the complex VQA task and leads to a promising performance improvement with a significantly reduced computation cost.

1 Introduction

Reasoning is critical for complex tasks and has attracted increasing attention from machine learning research. While data-driven methods, especially deep learning, have proven to work in an end-to-end fashion, the lack of explainability, high computational cost, and requirements of large amounts of data hinder their applications in the real world. In particular for VQA, data-driven VQA models are prone to exploit biases in datasets to find shortcuts instead of performing high-level reasoning [Kervadec *et al.*, 2021], and cannot maintain reasoning consistency in answering the compositional question and its sub-questions [Jing *et al.*, 2022]. To enhance learning efficiency and explainability, NS learning has been studied to combine the high explainability, provable correctness, and ease of using human expert knowledge of symbolic manipulation with the advantages of neural networks (NN).

However, NS methods cannot eliminate the disadvantages of NN. There is inevitable uncertainty in the NN part, due to probabilistic variations in random events or the lack of knowledge of a process. Most existing methods for NS VQA focus on reducing the requirements of symbolic labels (e.g., neural-symbolic concept learner (NS-CL)[Mao *et al.*, 2019]), learning new symbols (e.g., meta-concept learner[Han *et al.*, 2019]), increasing the complexity of tasks, (e.g., video question answering, requiring machines to understand physical laws[Chen *et al.*, 2021]), without considering the uncertainty propagation along the reasoning path. The absence of uncertainty awareness for reasoning fails to consider the long-tail distribution of visual concepts and the unequal importance of reasoning steps in real data [Li *et al.*, 2021], which can result in mistakes that are intolerable for safety-critical applications. For example, Shah *et al.* have shown that the existing VQA models are brittle to linguistic variations.

In this paper, we propose a Confidence-Based Neural-Symbolic (CBNS) approach that leverages uncertainty quantification of the deep learning models in the neural-symbolic system for confidence-based neural-symbolic VQA. Instantiated in the context of VQA, we consider the uncertainty in both scene perception and question parsing with Variational Dropout [Kingma *et al.*, 2015]. The uncertainty quantification can be utilized to improve the learning efficiency of the entire system and proves to be effective for confidence evaluations. Specifically, for the question parser, we introduce reconstruction loss, agreement loss, and variational dropout for training, and propose an improved beam search for inference, such that the question parser can achieve high accuracy with limited training data. Moreover, using the trained question parser with uncertainty quantification, we propose a data augmentation method to select predicted programs by the agreement loss and confidence evaluations. This proposed data augmentation method ensures a high probability of correct selection, allowing the selected programs to be used as pseudo groundtruth programs. Using pseudo groundtruth programs, we train the scene perception module without groundtruth annotations. In this way, our approach trains the VQA system with limited groundtruth concepts and programs, avoid-

ing the data-consuming and compute-intensive REINFORCE [Williams, 1992] used in prior work such as NS-CL[Mao *et al.*, 2019]. At the reasoning stage, the CBNS approach quantifies the uncertainties of both the question parser and scene perception modules to evaluate the confidence of the object concept predictions and the symbolic reasoning process. The concept predictions, along with their confidence evaluations, are input into the predicted symbolic programs for confidence-based reasoning.

Compared with the mainstream neural symbolic VQA, where only one representation for an image and one deterministic program for an associated question are predicted, and a single answer is provided at the end, our framework offers confidence evaluations for each step of the inference starting from the scene perception, which simplifies the error analysis process by highlighting the incorrectness of prediction with low confidence scores. To summarize, this paper makes the following contributions:

- We extend the neural symbolic learning by presenting the CBNS framework, which provides confidence evaluations for the NN modules in NS models. Additionally, CBNS is compatible with mainstream NS VQA architectures (such as NS-VQA and NS-CL);
- With the program prediction confidence estimated by CBNS, we significantly improve the learning efficiency of question parsing for current NS VQA approaches by avoiding REINFORCE, which is of high sample complexity and used to train question parser without the requirements of groundtruth programs;
- We evaluate the CBNS framework on the CLEVR datasets, and compare its performance with state-of-the-art neural symbolic VQA methods. Results show that CBNS is able to perform VQA tasks in a transparent way with accurate confidence estimation.

2 Related Works

Uncertainties in VQA have been studied to improve the performance of models. [Patro *et al.*, 2019] considered the uncertainty in vision, [Shah *et al.*, 2019] considered the uncertainty in questions, and [Vedantam *et al.*, 2019] proposed probabilistic neural symbolic models. However, few works explicitly consider the joint uncertainty from perception and reasoning. Imperfect computer vision interpretation and compositional question understanding introduce uncertainty that must be jointly reasoned to determine the correct answer[Krishnamurthy *et al.*, 2016]. In this paper, we aim to quantify the uncertainties in both scene perception and question parsing, make inferences based on uncertainty quantification, and provide answers with confidence evaluations.

Neural Semantic Parsing can be used to transform questions into explicit programs as sequences of symbolic tokens, which gain better interpretability than implicit programs as conditioned neural operations. Lots of methods have been developed to employ uncertainty quantification in natural language processing. [Dong *et al.*, 2018] designed metrics to quantify major causes of uncertainty and used the metrics to estimate confidence scores that indicate whether model predictions are likely to be correct. [Wang *et al.*, 2019] improved back-translation with the word- and sentence-level confidence estimation based on uncertainty. [Zhang *et al.*, 2019] proposed an adaptive decoding method that is guided by model uncertainty and automatically uses deeper computations when necessary. [Fomicheva *et al.*, 2020] exploited the machine translation model uncertainty to generate multiple diverse translations. In this paper, we are interested in exploiting the uncertainty in semantic parsing to generate multiple programs for the pure symbolic executor such that multiple reasoning paths can be investigated to improve the model performance.

3 CBNS Approach for VQA

The proposed CBNS approach for VQA shown in Figure 1 consists of three modules:

- A *scene perception* module to extract the object-based representations of images with confidence evaluations for each concept prediction (Section 4.2);
- A *question parser* module to translate a natural language question into multiple programs associated with confidence scores (Section 4.1). The multiple programs can facilitate finding accurate programs.
- A *program executor* module to execute the programs from the question parsing module on the concept quantization from the scene perception module with confidence evaluations for answer predictions (Section 4.3).

One challenge of NS VQA is to reduce or even avoid the requirements of groundtruth programs for training the question parser. The authors [Mao *et al.*, 2019; Han *et al.*, 2019] employed REINFORCE for the optimization of the question parser in a non-smooth program space, which requires the correctness of the execution result as the reward signal. However, REINFORCE suffers from an inefficient update process and noisy gradient estimate, which negatively affects training the scene perception module. Instead, we propose to use semi-supervised learning to avoid REINFORCE and improve learning efficiency. The idea is to learn a sufficiently accurate question parser with uncertainty quantification from limited fully-annotated data, which will be elaborated on next.

4 CBNS Model Training and Inference

The training procedure of the CBNS model includes: (1) training the question parser using limited fully-annotated data including scene annotations, questions, programs, and answers; (2) predicting programs for the questions sampled from the training set including only images, questions and answers, and then selecting programs based on the proposed data augmentation method; (3) training the concept learner using the selected programs as pseudo groundtruth programs and the associated images and answers.

4.1 Uncertainty-aware Question Parser Training

Similar to [Yi *et al.*, 2018], we use an attention-based sequence-to-sequence (seq2seq) model with an encoder-decoder structure as a question parser to transform questions

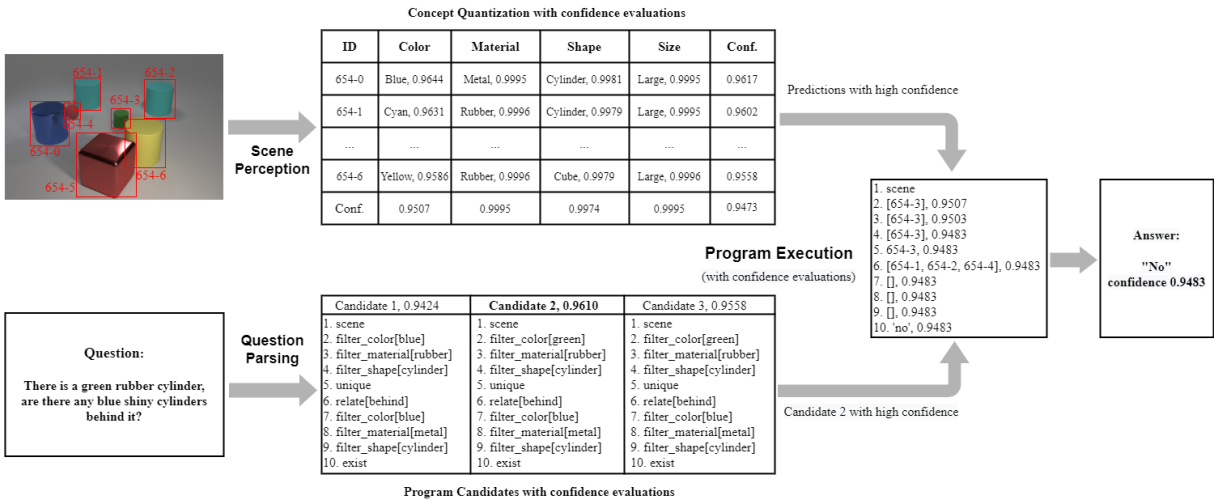

Figure 1: The schematic of CBNS VQA. The model consists of a question parsing module, a scene perception module, and a program execution module.

into symbolic programs. Specifically, the encoder is represented by a bidirectional LSTM [Hochreiter and Schmidhuber, 1997] that takes as input a question of variable lengths and outputs an encoded vector $e_i = [e_i^F, e_i^B]$ at time step i by

$$\begin{aligned} e_i^F, h_i^F &= \text{LSTM}(\Phi_E(x_i), h_{i-1}^F), \\ e_i^B, h_i^B &= \text{LSTM}(\Phi_E(x_i), h_{i+1}^B), \end{aligned} \tag{1}$$

where Φ_E is the jointly trained word embedding for the encoder; (e_i^F, h_i^F) and (e_i^B, h_i^B) are the outputs and hidden states of the forward and backward networks, respectively. The decoder is a similar LSTM. The output $q_t^P, s_t = \text{LSTM}(\Phi_D(y_{t-1}), s_{t-1})$ of the LSTM, where y_{t-1} denotes the previous token of the output sequence and Φ_D is the decoder word embedding, is then fed to an attention layer with identity attention matrix to obtain a context vector c_t^P as a weighted sum of the encoded states e_i via

$$\alpha_{ti}^P = \text{softmax}(q_t^{P\top} e_i), \; c_t^P = \sum_i \alpha_{ti}^P e_i. \tag{2}$$

Then, $[q_t^P, c_t^P]$ is passed to a fully connected layer with the softmax activation function to obtain the conditional distribution of the predicted token $\hat{y}_t$.

To enhance the performance of the question parser, we improve the architecture of the question parser model. We add a reconstructor [Tu *et al.*, 2016], which reconstructs the question from the hidden layer of the decoder and ensures that the program retains the information in the question, to the question parser model [Yi *et al.*, 2018]. The reconstructor is a similar decoder. The output $q_i^R = \text{LSTM}(\Phi_R(x_{i-1}))$ of the LSTM is then fed to an attention layer as

$$\alpha_{it}^R = \text{softmax}(q_i^{R\top} W_A s_t), \; c_i^R = \sum_t \alpha_{it}^R s_t, \tag{3}$$

where W_A is the attention weight matrix. We obtain the distribution for the predicted token by $x_i^R \sim \text{softmax}(W_O^R[q_i^R, c_i^R])$. Then, the reconstruction loss is

$$R(\mathbf{x}^n|\mathbf{s}^n;\gamma) = \frac{1}{N}\sum_{n=1}^{N} \log p(\mathbf{x}^n|\mathbf{s}^n;\gamma), \tag{4}$$

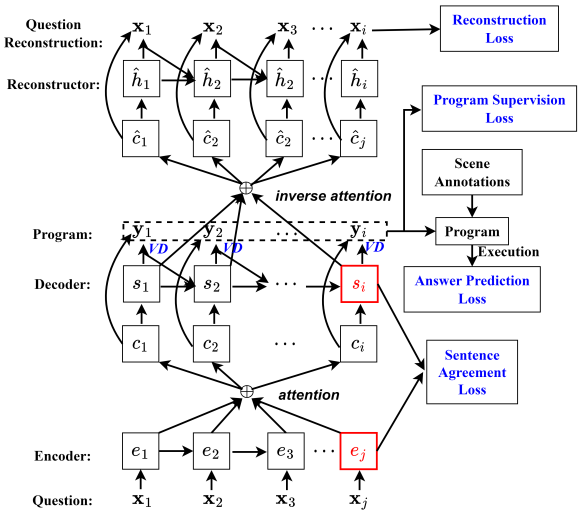

Figure 2: Architecture of our question parser.

where $\mathbf{s}^n$ denotes the sequence of the hidden states of the decoder for the n-th question. However, neither the seq2seq model nor the reconstructor can guarantee sequence level agreement [Yang *et al.*, 2019], as the predictions of the seq2seq model and the reconstructor at the current time step are based on the predictions at the previous time step. To enforce the sequence level agreement, we add a sequence agreement loss as follows

$$A(\mathbf{x}^n, \mathbf{y}^n) = \frac{1}{N}\sum_{n=1}^{N} \|e_E(\mathbf{x}^n) - s_D(\mathbf{y}^n)\|^2 \tag{5}$$

where $e_E \in \mathbb{R}^d$ and $s_D \in \mathbb{R}^d$ denote the hidden states of the encoder and decoder at the last time step, respectively, and d is the dimension of the hidden states.

Uncertain quantification in Question Parsing

Bayesian posterior inference is a commonly used method for uncertainty quantification [Bao *et al.*, 2021]. While exact in-

ference is computationally intractable on account of the high complexity of NN, efficient approximate schemes such as Markov Chain Monte Carlo and variational inference can be designed [Kingma *et al.*, 2015]. Variational dropout (VD) has proven to greatly improve the efficiency of variational Bayesian inference on the model parameters [Kingma *et al.*, 2015], which is critical for quantifying uncertainty throughout the complex VQA reasoning process. Therefore, we use VD [Kingma *et al.*, 2015] to quantify the model uncertainty.

In particular, similar to [Kingma *et al.*, 2015], we assume the scale-invariant log-uniform prior $p(\mathbf{w})$ and a factorized Gaussian posterior $q_\phi(\mathbf{w})$ with trainable parameters ϕ for the weights $\mathbf{w}$ of a hidden/output layer the NN. To approximate the posterior of $\mathbf{w}$ by $q_\phi(\mathbf{w})$, ϕ is learned by maximizing

$$\mathcal{L}(\phi) = -D_{KL}(q_\phi(\mathbf{w}) \| p(\mathbf{w}))) + L_{\mathcal{D}}(\phi), \tag{6}$$

where $L_{\mathcal{D}}(\phi) = \sum_{n=1}^{N} \mathbb{E}_{q_\phi(\mathbf{w})}[\log p(\mathbf{y}^n|\mathbf{x}^n, \mathbf{w})]$ and is approximated by an unbiased differentiable minibatch-based Monte Carlo estimator

$$L_{\mathcal{D}}(\phi) \approx \sum_{n=1}^{N} \left(\frac{1}{M} \sum_{k=1}^{M} \log p(\mathbf{y}^n|\mathbf{x}^n, \mathbf{w}^{(k)}) \right) \tag{7}$$

with $\mathbf{w}^{(k)}$ sampled from $q_\phi(\mathbf{w})$. In particular, $q_\phi(\mathbf{w}_i) = \mathcal{N}(\phi_i, \alpha\phi_i^2)$, and $\mathbf{w}_i = \phi_i + \sqrt{\alpha\phi_i^2}\epsilon_i$ where $\mathbf{w}_i$ denotes the i-th weight, $\epsilon_i \sim \mathcal{N}(0, 1)$, and α is the dropout rate. We incorporate variational dropout in the decoder of our question parser for uncertainty quantification.Specifically, using the decoder output q_t^P and the context vector c_t^P, we obtain the distribution for the predicted token by

$$\hat{y}_t \sim \text{softmax}(W_O \text{ReLU}(\text{VD}([q_t^P, c_t^P]; \mathbf{w}))),$$

where $\text{VD}(\cdot; \mathbf{w})$ denotes the NN layer where variational dropout is applied to approximate the posterior of $\mathbf{w}$ for uncertainty quantification. A deterministic weight matrix W_O is used for token prediction to reduce the model complexity.

Figure 2 shows the complete architecture of our question parser. The program supervision loss and the answer prediction loss are borrowed from [Yi *et al.*, 2018]. The program supervision loss requires groundtruth programs. The answer prediction loss is computed based on the difference between the groundtruth answers and the answers that are predicted by executing the predicted programs on the concepts from the scene perception module.

Uncertainty-aware Inference of Question Parser

Using Monte Carlo (MC) sampling by $q_\phi(\mathbf{w})$, we obtain multiple models and use model averaging for program generation. Specifically, we sample M outputs $\{h^{(k)} = W_O \text{ReLU}(\text{VD}([q_t^P, c_t^P]; \mathbf{w}^{(k)}))\}_{k=1}^{M}$, and use the outputs to estimate the conditional distribution of the token y_t as

$$\hat{p}(y_t|y_{t-1}, \cdots, y_1, x_i) = \text{softmax}(\frac{1}{M} \sum_{k=1}^{M} h^{(k)}). \tag{8}$$

Furthermore, to exploit and deal with the uncertainty in the question parser learned from data, we generate multiple programs. Beam search (BS) is a widely-used test-time decoding algorithm in neural machine translation, but it suffers from a lack of diversity. Approaches have been proposed to enhance diversity[Vijayakumar *et al.*, 2016; Li and Jurafsky, 2016]. However, the diversity of BS is determined by $\hat{p}$. When $\hat{p}$ is close to a uniform distribution, then BS can generate diverse sequences; if $\hat{p}$ is close to one-hot encoding, simply enforcing diversity can increase the discrepancy between the training and testing process of BS and thus degrade the performance of decoding. Moreover, minimizing the negative log-likelihood in the training process is prone to result in overly confident predictions and uncalibrated uncertainty which does not correspond well with the model error [Laves *et al.*, 2020]. Therefore, in this paper, we introduce variational dropout to moderate this problem by considering model uncertainty. To generate B programs, at each time step t of decoding, we store the top-B beam candidates of symbolic modules, where B is the *beam width* and the candidates are sorted by $\Theta(y_t) = \sum_{i=1}^{t} \theta(y_i)$ with $\theta(y_i) = \log \hat{p}(y_i|y_{i-1}, \cdots, y_1, \mathbf{x}^n)$ At the next time step, we consider all possible single token extensions of these beams and selects the B most likely extensions. This process is repeated until maximum time T. Then the most likely B sequences $\{\hat{\mathbf{y}}^{(b)}\}_{b=1}^{B}$ are selected.

However, the log-probabilities of sequences may not align with the probability that the program candidates are correct due to the model error/uncertainty. Instead, we consider model uncertainty to determine the most promising program. In particular, we calibrate the probabilities of sequences by penalizing the average estimations with variances as

$$\hat{p}_c(\hat{\mathbf{y}}^{(b)}) = \frac{\mathbb{E}\left[\hat{p}(\mathbf{y}^{(b)}|\mathbf{x})\right]}{\text{Var}(\hat{p}(\mathbf{y}^{(b)}|\mathbf{x}))}. \tag{9}$$

where $\mathbb{E}\left[\hat{p}(\mathbf{y}^{(b)}|\mathbf{x})\right] = \frac{1}{M}\sum_{k=1}^{M} \hat{p}^{(k)}(\mathbf{y}^{(k,b)}|\mathbf{x}, \mathbf{w}^{(k)})$ and $\text{Var}(\hat{p}(\mathbf{y}^{(b)}|\mathbf{x})) = \frac{1}{M}\sum_{k=1}^{M} \left(\hat{p}^{(k)}(\mathbf{y}^{(k,b)}|\mathbf{x}, \mathbf{w}^{(k)}) - \mathbb{E}\left[\hat{p}(\mathbf{y}^{(b)}|\mathbf{x})\right]\right)^2$ with $\hat{p}^{(k)}(\mathbf{y}^{(k,b)}|\mathbf{x}, \mathbf{w}^{(k)})$ being computed using $\mathbf{w}^{(k)}$ sampled by the posterior distribution. Then, we sort the B sequences by $\hat{p}_c(\hat{\mathbf{y}}^{(b)})$ rather than using $\Theta(y_T^{(b)})$ as BS.

Confidence Scores of Program Candidates

To enable the aggregation of the uncertainty quantification of individual modules for confidence-based reasoning, we need a confidence score for the generated program candidates based on the uncertainty of predictions. In this paper, we evaluate the confidence score c_s of the model by

$$c_s(\hat{p}(\mathbf{y}|\mathbf{x})) = \left(1 - \frac{\text{Var}(\hat{p}(\mathbf{y}|\mathbf{x}))}{\mathbb{E}\left[\hat{p}(\mathbf{y}|\mathbf{x})\right]}\right)^{\beta} \in [0, 1], \tag{10}$$

where the values of $\hat{p}(\mathbf{y}|\mathbf{x})$ are estimated using the weights drawn from $q_\phi(\mathbf{w})$ using the MC sampling, β is a tuning hyper-parameter to control the difference between confidence values of programs of larger differences. $c_s(\hat{p}(\mathbf{y}|\mathbf{x}))$ is between 0 and 1, as the variance of a probability is no greater than the corresponding expectation [Wang *et al.*, 2019]. Moreover, $c_s(\hat{p}(\mathbf{y}|\mathbf{x}))$ increases as the uncertainty (described by variances) decreases, and $c_s(\hat{p}(\mathbf{y}|\mathbf{x}))$ is positively correlated with the calibrated probability $\hat{p}_c(\mathbf{y})$ defined in Eq. (9).

Data Augmentation Method for Program Selection

The uncertainty-aware training of the question parser can also be used for data augmentation to avoid the requirements of

Algorithm 1 Program selection by data augmentation rules

Input: questions $\{\mathbf{x}^n\}_{n=1}^N$, required program size N_a
for $n = 1$ **to** N **do**
  for $k = 1$ **to** M **do**
    Sample a model $\hat{p}^{(k)}$ by the posterior $q_\phi(\mathbf{w})$
    Generate B program candidates $\hat{\mathbf{y}}^{(k,b)}$ by $\hat{p}^{(k)}$ and save $\hat{p}^{(k)}(\hat{\mathbf{y}}^{(k,b)}|\mathbf{x}^n)$, $b = 1, \cdots, B$
  end for
  Generate B program candidates $\hat{\mathbf{y}}^{(b)}$ by $\hat{p}$ in Eq. (8)
  Compute the calibrated probability by Eq. (9)
  for $b = 1$ **to** B **do**
    Compute the agreement loss $A(\mathbf{x}^n, \hat{\mathbf{y}}^{(k,b)})$ by Eq. (5)
  end for
  Obtain the ranking rank^A of the B candidates $\{\hat{\mathbf{y}}^{(b)}\}_{b=1}^B$ by the agreement loss
  Obtain the ranking rank$^{\hat{p}_c}$ of $\{\hat{\mathbf{y}}^{(b)}\}_{b=1}^B$ by the calibrated probabilities using $\{\hat{p}^{(k)}(\hat{\mathbf{y}}^{(k,b)}|\mathbf{x}^n)\}_{k,b=1,1}^{M,B}$
  if Top-1@rank^A==Top-1@rank$^{\hat{p}_c}$ **then**
    Select the Top-1 program of rank^A and save the calibrated probability of the selected program
  end if
end for
Sort the selected programs by the calibrated probability
Return: Top N_a programs and the associated questions.

groundtruth programs or using the programs from the exploration of REINFORCE to train the concept learner, which could greatly increase the training/data efficiency of concept learner. In particular, we propose to select programs that are correct with a high probability from the predicted programs of the questions in the training set for concept learning.

To increase the probability that the selected programs are correct, we select programs on which multiple rankings reach consensus. Specifically, after obtaining B programs from BS, we rank the programs by the calibrated probabilities; then, we obtain another ranking by the agreement loss between the candidates and the question, as the agreement loss can measure the coverage of the programs to the questions.

After obtaining the two rankings, we select the questions when the two rankings reach a consensus on the top-1 programs. Next, we rank the selected questions again by the calibrated probabilities of the top-1 program and augment the dataset with the top questions associated with the top-1 predicted programs. The procedures of the program selection for data augmentation are summarized in Algorithm 1.

4.2 Concept Learner Using Predicted Programs

Since the accuracy of the selected programs by the data augmentation rules is high, we use the selected programs as pseudo groundtruth programs to learn the parameters of the scene perception module. To quantify the uncertainty in scene perception module, we again apply Variational Dropout to the object features.

For determining an object's concepts (concept quantization), similar to [Mao *et al.*, 2019], we use a neural operator that maps the object representation to an embedding. Then, the attribute is determined based on the cosine distances between the learned concept vectors v and the embedding of the object. For example, the probability of the concepts that belong to attribute a_p for an object o_p is estimated by

$$\hat{p}_{a_p}(c_p|o_p; u) = \sigma\left(b^{a_p} + \frac{< \|u(o_p)\|_2, v^{c_p} > -1 + \gamma}{\gamma\tau}\right),$$

where b^{a_p} is a trainable log-softmax-normalized vector that indicates whether c_p belongs to a_p, $u(\cdot)$ denotes the neural operator, and v^{c_p} is the L2-normalized concept vector of the concept c_p; σ denotes the softmax function and $< \cdot, \cdot >$ denotes the cosine distance; γ and τ are scalar constants for scaling and shifting the values of similarities.

In the training phase, for each batch of data, we randomly sample one model $u^{(k)}(\cdot)$ (with k denoting the index of the sampled model) by the dropout rate, and compute one embedding $u^{(k)}(o_p)$ for concept quantization of the object o_p. The optimization objective of the scene perception module is to maximize the likelihood of the final answers a^n being correct

$$\max_{\vartheta} \sum_{n=1}^{N} \sum_{k=1}^{M} \ell(a^n, \mathrm{E}(\mathrm{P}(i_p^n; \vartheta^{(k)}), \hat{P}^n)), \tag{11}$$

where E is the executor, and P the scene perception module with parameters $\vartheta^{(k)}$; a denotes the answer, i_p the image, and $\hat{P}^n$ the pseudo groundtruth program from the candidates for the n-th question. To make the execution outputs of $\hat{P}^n$ fully differentiable w.r.t. the parameters in the scene perception module for concept learning, we use a quasi-symbolic executor [Mao *et al.*, 2019; Han *et al.*, 2019], i.e., the intermediate results of the programs are represented as the attention mask over all objects in the scene. Each element of the mask $Mask_i \in [0, 1]$ denotes the probability that the i-th object of the scene belongs to the intermediate results.

For concept quantization with confidence evaluation, we compute M embeddings $\{u^{(k)}(o_p)\}_{k=1}^M$ of the object o_p by sampling M models, and compute the probabilities $\hat{p}_{a_p}^{(k)}(c_p|o_p)$ w.r.t. to all the concepts $c_p \in C_p$ for each embedding $u^{(k)}(o_p)$. Then, we compute the confidence scores

$$c_s(\hat{p}_{a_p}(c_p|o_p)) = \left(1 - \frac{\mathrm{Var}(\hat{p}_{a_p}(c_p|o_p))}{\mathbb{E}\left[\hat{p}_{a_p}(c_p|o_p)\right]}\right)^2. \tag{12}$$

To calibrate $\hat{p}_{a_p}(c_p|o_p)$ by uncertainty, we use $c_s(\hat{p}_{a_p}(c_p|o_p))$ to weight $\hat{p}_{a_p}(c_p|o_p; \bar{u})$ calculated by the average embeddings $\bar{u}(o_p) := \frac{1}{M}\sum_{k=1}^M u^{(k)}(o_p)$, and the weighted probabilities $\bar{\hat{p}}_{a_p}(c_p|o_p) := c_s(\hat{p}_{a_p}(c_p|o_p))\hat{p}_{a_p}(c_p|o_p; \bar{u})$ are used for concept quantization. The prediction of the attribute value is the concept c_p^* with the highest weighted probability, i.e., $c_p^* = \arg\max_{c_p} \bar{\hat{p}}_{a_p}(c_p|o_p)$.

Moreover, we use the average confidence score

$$c_s(a_p|o_p) = \frac{1}{|C_p|} \sum_{c_p \in C_p} c_s(\hat{p}_{a_p}(c_p|o_p))$$

as the confidence score of an attribute a_p for an object o_p, and use the minimal value of the confidence scores of an attribute for all objects in an image as the confidence score of

an attribute of the image, i.e., $c_s(a_p|i_p) = \min_{o_p} c_s(a_p|o_p)$. Then, we use the products of the confidence scores of all the attributes as the confidence score of the image, i.e., $c_s(i_p) = \prod_{a_p} c_s(a_p|i_p))$.

4.3 Confidence Scores throughout Execution

After concept quantization of an image and program generation of a question about the image with confidence evaluations, we evaluate the confidence for each step of the program execution. Specifically, for the t-th functional operation in the program that involves an attribute a_p, we use

$$c_s^t = \min_{o_p \in O_t} c_s(a_p|o_p) \tag{13}$$

where O_t denotes the set of objects that are involved in the t-th operation, as the confidence score of the t-th step. Then, we compute the confidence score of the answer derived by executing the program as

$$c_s(i_p, \mathbf{x}, \mathbf{y}) = \left(\sqrt[T]{\prod c_s^t} \right)^{\alpha} \times c_s(\hat{p}(\mathbf{y}|\mathbf{x}))^{1-\alpha} \tag{14}$$

where T is the number of operations involving attributes in the program, $\sqrt[T]{\cdot}$ is used to normalize the score, and α is a tuning parameter to control the relative importance of the final confidence scores of perception and program. We choose α to achieve the largest AUC score of using the answer confidence score to predict the correctness of answers on the training set.

5 Experiments

In this section, we empirically evaluate the proposed framework CBNS VQA on the CLEVR dataset [Johnson *et al.*, 2017a]. The goal is to evaluate whether CBNS VQA is capable of providing confidence-based inference for VQA tasks and how it performs compared with state-of-the-art neural symbolic VQA methods.

5.1 Module training of CBNS model

Question parser training details: We train the question parser using 630 ground-truth programs (7 programs per question template, $< 1\%$ of CLEVR's 70K training images) with the associated groundtruth scene annotations from the training set of CLEVR dataset. First, we use the groundtruth scene annotations and programs to train the question parser in Section 4. Our question parser shares the same encoder and decoder structures as [Yi *et al.*, 2018] except for the reconstructor, the variational dropout layer, and the additional losses for self-supervision. We trained our question parser for 20k iterations.

Table 1: Answer accuracy of question parser models for different numbers of supervised question program examples.

#	270	450	630	810
MLE	71.97	83.35	90.22	92.17
OURS	94.11	97.84	99.00	99.35

Performance comparison of question parsing: Table 1 shows the reconstruction loss, agreement loss, and VD with BS improved the answer accuracy by 8.78%, compared against the maximum likelihood estimation (MLE) question parser from [Yi *et al.*, 2018]. The beam width $B = 3$ in our experiments. Moreover, we validate the data augmentation rules by checking the accuracy of the selected programs. Specifically, we use the exact match score (i.e., the percentage of predicted programs that exactly match the groundtruth programs[1]) to measure the accuracy of the predicted programs. The accuracy of the predicted programs selected by the rules on the validation set is 99.88% while the accuracy of all the predicted program candidates is 99.00%, which demonstrates the high effectiveness of the rules. Moreover, our question parser improves the program accuracy by more than 5%, compared against NMN[Johnson *et al.*, 2017b] (62.47%) and Prob-NMN [Vedantam *et al.*, 2019] (93.15%) using 1000 supervised program question examples. Additionally, our question parser significantly reduced the computational cost, compared to the question parser in the NS-VQA. NS-VQA question parser used 270 groundtruth programs for 20k-iteration supervised pretraining and then ran additional 2M iterations of REINFORCE with early stopping on 30k data while our method does not need REINFORCE at all.

Confidence evaluation of question parsing: To demonstrate the effectiveness of confidence evaluations, We test the capability of the confidence scores of programs to predict the correctness of the programs. The AUC score of using the confidence scores to predict the correctness of the programs that reach consensus on the Top 1 ranking is 0.9450, which shows the usefulness of the confidence score for predicting the program's correctness.

Scene perception training details: We generate 3 program candidates for each of the 90k questions randomly selected from the CLEVR training set and find 64k questions for which the Top-1 program candidates by the rankings of agreement loss and the calibrated probabilities reach consensus. Then, we sort the Top-1 program candidates for the 64k questions by the calibrated probabilities in descending order and select the first 45k programs associated with the images and answers for training the concept learner. Our scene perception module shares the same structure as NS-CL except for the variational dropout layer. To train the scene perception module, we use a similar approach as NS-CL except that we use the fixed programs predicted by our question parser while NS-CL trains the question parser (using REINFORCE and additional data) and the scene perception module alternatively. It is noted that REINFORCE suffers from an inefficient update process and noisy gradient estimate, which negatively affects the scene perception module training. Our approach can avoid REINROECE and achieve competitive accuracy for both question parsing and scene perception.

Performance comparison of scene perception: We evaluate the performance of our scene perception module on the CLEVR validation set, and the accuracy for all object properties is greater than 98%, suggesting the predicted programs are sufficient for concept learning without direct concept supervision. Moreover, our scene perception module achieved comparative accuracy $(99.32)\%$ to the NS-CL on the training set of CLEVR dataset with groundtruth programs (99.90%)

[1]It is noted that the groundtruth programs are only used for validation rather than training or inference.

Table 2: Answer accuracy comparison for different question types. RL refers to REINFORCE; GT: groundtruth; L: using concept labels; PS: pure symbolic; QS: quasi-symbolic.

Model	Samples	Prog.	RL	Exe.	Mean	Count	Cmp. Num.	Exist	Query	Cmp. Attr.
NS-CL	30k	-	-	PS	96.33	96.70	97.27	97.92	95.80	95.24
	70k	0	Yes	QS	98.19	96.31	98.06	98.94	98.80	98.84
CBNS(CL)	45k	630	**No**	PS	96.11	94.66	97.55	97.01	96.09	96.67
	70k	630	**No**	PS	**98.36**	99.75	99.00	98.35	98.41	99.18
NS-VQA	30k L	270	Yes	PS	99.13	99.12	99.45	99.43	99.06	98.90
CBNS(VQA)	30k L	630	**No**	PS	98.46	97.40	99.10	98.40	98.58	99.34

regarding the mean concept accuracy, while also providing reliable confidence evaluations.

Confidence evaluation of scene perception: In the concept quantization from CBNS(CL), we use the confidence scores to predict the correctness of predicted concepts and calculate the AUC scores to evaluate the confidence's performance. Specifically, the AUC scores for the attributes *color*, *material*, *shape*, and *size* are 0.9768, 0.9652, 0.9670 and 0.9908, which demonstrates the effectiveness of the confidence evaluations of concept learning.

5.2 Final performance comparison

Table 2 shows the accuracy comparison between our approach and the existing works. Need to mention, except for our proposed CBNS, all other neural-symbolic VQA systems listed here do not provide confidence evaluation for the reasoning process, they only treat the intermediate results as black-box outcomes.

Since NS-CL avoids the requirements of concept labels with quasi-symbolic reasoning and CNBS uses pure symbolic reasoning, for a fair comparison, we test the performance of concept quantization from NS-CL with pure symbolic reasoning. Utilizing the groundtruth programs for symbolic execution, NS-CL achieved an accuracy of 96.33%. Our proposed method CBNS(CL) achieved a similar performance (96.11%) to NS-CL using pure symbolic reasoning but avoided the REINFORCE training process of high sample complexity. Moreover, the accuracy of our approach can further increase by improving the accuracy of concept quantization. Training with 70K samples for concept quantization, our approach (98.36%) performs better than NS-CL (98.19%) using quasi-symbolic reasoning.

Additionally, to demonstrate our CBNS framework is model-agnostic, we apply our confidence-based approaches to NS-VQA [Yi *et al.*, 2018] and compare the accuracy of NS-VQA[2] with our CBNS(VQA) that uses the proposed question parser with confidence evaluations. The concept labels were used to train the attribute net. We used the same predictions of concepts for evaluating the NS-VQA parser and our parser. The answer accuracy of NS-VQA (99.13%) is the same as that of using the groundtruth programs and slightly higher than CBNS(VQA) (98.46%), as the question parser of NS-VQA was pretrained on 270 groundtruth question program pairs and finetuned on $30,000$ question-answer pairs while our parser was trained only on 630 fully annotated data.

Confidence evaluation of answers: Using the similar settings of NS-CL[3] to train the perception module but with uncertainty quantification, we evaluated the confidence of the logic operation outputs along the program execution trace, as described in Section 4.3. In Figure 3, we compare the ROC curves of CBNS VQA with another two baselines: (1) MAC model + Predictive Uncertainty; and (2) MAC model + EDL. The MAC [Hudson and Manning, 2018] is an end-to-end neural structure that performs reasoning implicitly and has shown good performance on VQA tasks. Predictive Uncertainty is introduced by [Malinin and Gales, 2018], it models the distributional uncertainty by parameterizing a prior distribution over predictive distributions. The other method, EDL[Sensoy *et al.*, 2018] uses subjective logic to measure the uncertainty of the prediction result. In the experiment, we noticed a significant performance drop after enabling the uncertainty measure on neural networks. The original validation accuracy of MAC on CLEVR, if combined with predictive uncertainty and EDL, drops from 98.9% to 95.36% and 90.67%, respectively. Instead, our CBNS(VQA) could maintain a good accuracy of 98.46% while getting the best uncertainty estimation with an AUC of 0.9499.

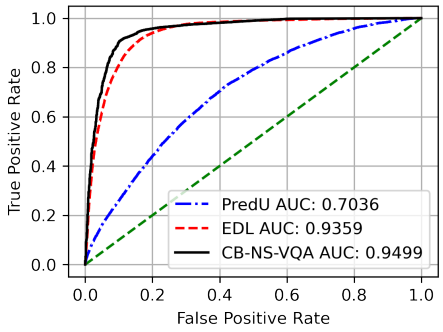

Figure 3: The ROC curves of answer correctness prediction for different uncertainty quantization methods.

6 Conclusion

In this paper, a CBNS framework for VQA is introduced, wherein confidence evaluations relied on the quantification of model uncertainty in the NN components (specifically, scene perception and question parsing) as well as the program execution. The system considers multiple program candidates and provides answers accompanied by confidence evaluations. Through experiments conducted on the CLEVR dataset, it is observed that uncertainty quantification not only serves as a basis for confidence assessments but also enhances the accuracy of question parsing and improves learning efficiency. Moving forward, future research will focus on exploring the adaptability of this framework to other neural-symbolic VQA models and testing its performance using larger-scale datasets of natural images and language.

[2]https://github.com/kexinyi/ns-vqa.

[3]https://github.com/vacancy/NS-CL-PyTorch-Release.

References

[Bao *et al.*, 2021] Yajie Bao, Javad Mohammadpour Velni, and Mahdi Shahbakhti. Epistemic uncertainty quantification in state-space lpv model identification using bayesian neural networks. *IEEE Control Systems Letters*, 5(2):719–724, 2021.

[Chen *et al.*, 2021] Zhenfang Chen, Jiayuan Mao, Jiajun Wu, Kwan-Yee Kenneth Wong, Joshua B Tenenbaum, and Chuang Gan. Grounding physical concepts of objects and events through dynamic visual reasoning. *arXiv preprint arXiv:2103.16564*, 2021.

[Dong *et al.*, 2018] Li Dong, Chris Quirk, and Maria Lapata. Confidence modeling for neural semantic parsing. In *56th Annual Meeting of the Association for Computational Linguistics*, pages 743–753. Association for Computational Linguistics, 2018.

[Fomicheva *et al.*, 2020] Marina Fomicheva, Lucia Specia, and Francisco Guzmán. Multi-hypothesis machine translation evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1218–1232. Association for Computational Linguistics, 2020.

[Han *et al.*, 2019] Chi Han, Jiayuan Mao, Chuang Gan, Josh Tenenbaum, and Jiajun Wu. Visual concept-metaconcept learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Hudson and Manning, 2018] Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067*, 2018.

[Jing *et al.*, 2022] Chenchen Jing, Yunde Jia, Yuwei Wu, Xinyu Liu, and Qi Wu. Maintaining reasoning consistency in compositional visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5099–5108, 2022.

[Johnson *et al.*, 2017a] 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*, pages 2901–2910, 2017.

[Johnson *et al.*, 2017b] 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*, pages 2989–2998, 2017.

[Kervadec *et al.*, 2021] Corentin Kervadec, Theo Jaunet, Grigory Antipov, Moez Baccouche, Romain Vuillemot, and Christian Wolf. How transferable are reasoning patterns in vqa? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4207–4216, 2021.

[Kingma *et al.*, 2015] Diederik P. Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick, 2015.

[Krishnamurthy *et al.*, 2016] Jayant Krishnamurthy, Oyvind Tafjord, and Aniruddha Kembhavi. Semantic parsing to probabilistic programs for situated question answering. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 160–170, 2016.

[Laves *et al.*, 2020] Max-Heinrich Laves, Sontje Ihler, Karl-Philipp Kortmann, and Tobias Ortmaier. Calibration of model uncertainty for dropout variational inference. *arXiv preprint arXiv:2006.11584*, 2020.

[Li and Jurafsky, 2016] Jiwei Li and Dan Jurafsky. Mutual information and diverse decoding improve neural machine translation. *arXiv preprint arXiv:1601.00372*, 2016.

[Li *et al.*, 2021] Zhuowan Li, Elias Stengel-Eskin, Yixiao Zhang, Cihang Xie, Quan Hung Tran, Benjamin Van Durme, and Alan Yuille. Calibrating concepts and operations: Towards symbolic reasoning on real images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14910–14919, 2021.

[Malinin and Gales, 2018] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31, 2018.

[Mao *et al.*, 2019] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*, 2019.

[Patro *et al.*, 2019] Badri N Patro, Mayank Lunayach, Shivansh Patel, and Vinay P Namboodiri. U-cam: Visual explanation using uncertainty based class activation maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7444–7453, 2019.

[Sensoy *et al.*, 2018] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018.

[Shah *et al.*, 2019] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-consistency for robust visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6649–6658, 2019.

[Tu *et al.*, 2016] Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. Neural machine translation with reconstruction, 2016.

[Vedantam *et al.*, 2019] Ramakrishna Vedantam, Karan Desai, Stefan Lee, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Probabilistic neural symbolic models for interpretable visual question answering. In *International Conference on Machine Learning*, pages 6428–6437. PMLR, 2019.

[Vijayakumar *et al.*, 2016] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016.

[Wang *et al.*, 2019] Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. Improving back-translation with uncertainty-based confidence estimation. *arXiv preprint arXiv:1909.00157*, 2019.

[Williams, 1992] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, pages 5–32, 1992.

[Yang *et al.*, 2019] Mingming Yang, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, Min Zhang, and Tiejun Zhao. Sentence-level agreement for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3076–3082, 2019.

[Yi *et al.*, 2018] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in neural information processing systems*, 31, 2018.

[Zhang *et al.*, 2019] Xiang Zhang, Shizhu He, Kang Liu, and Jun Zhao. Adansp: Uncertainty-driven adaptive decoding in neural semantic parsing. In *ACL*, 2019.