[74] Q. Wu, Z. Lan, K. Qian, J. Gu, A. Geramifard, and Z. Yu. Memformer: A memory-augmented transformer for sequence modeling. In Y. He, H. Ji, S. Li, Y. Liu, and C.-H. Chang, editors, *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 308–318, Online only, Nov. 2022. Association for Computational Linguistics.

[75] Y. Wu, Y. Zhao, B. Hu, P. Minervini, P. Stenetorp, and S. Riedel. An efficient memory-augmented transformer for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2210.16773*, 2022.

[76] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.

[77] K. Yi, C. Gan, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*, 2019.

[78] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in neural information processing systems*, 31, 2018.

[79] S. Yu, J. Cho, P. Yadav, and M. Bansal. Self-chained image-language model for video localization and question answering. *arXiv preprint arXiv:2305.06988*, 2023.

[80] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6281–6290, 2019.

[81] P. Zhang, X. Li, X. Hu, J. Yang, L. Zhang, L. Wang, Y. Choi, and J. Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5579–5588, 2021.

A Appendix / supplemental material

B Elaborated Experiments and Results Discussion

B.1 STAR

We provide results on the STAR Test, further baselines and model ablations for video reasoning tasks in table 5.

Table 5: **Left:** Results on STAR [72] official hidden test set (evaluation server) with ground-truth vision (GT V) and predicted vision (PR V); **Right:** Results on STAR val. set with num. of sampled frames =32 unless otherwise stated in (); IPRM outperforms prior state-of-art SeviLA-BLIP2 VLM across question types.

Model	Setup	STAR-Test				
		Int.	Seq.	Pred.	Feas.	Avg.
Vis-BERT[43]	GT V	34.7	35.9	31.2	31.4	34.7
CLIP-BERT[38]	GT V	36.3	38.9	30.7	29.8	36.5
NS-SR[72]	GT V	42.6	46.3	43.4	43.9	44.5
IPRM	GT V	**70.5**	**83.8**	**85.3**	**79.1**	**79.7**
Vis-BERT[43]	-	33.6	37.2	31.0	30.8	34.8
CLIP-BERT[38]	-	39.8	43.6	32.2	31.4	36.7
NS-SR[72]	PR V	30.9	31.8	30.2	29.7	30.7
SHG-VQA [62]	-	48.0	42.0	35.3	32.5	39.5
GF [3]	-	56.1	61.3	52.7	45.7	53.9
mPLUG [40]	-	60.4	65.6	57.5	49.6	58.3
IPRM	PR V	**61.7**	**72.7**	**75.4**	**71.3**	**70.3**

Model	Int.	Seq.	Pred.	Feas.	Avg.
All-in-One [66]	47.5	50.8	47.7	44.0	47.5
Temp[ATP](32) [5]	50.6	52.8	49.3	40.6	48.3
MIST [16]	55.5	54.2	54.2	44.4	51.1
InternVideo(8) [69]	62.7	65.6	54.9	51.9	58.7
SeViLA-BLIP2 [79]	63.7	70.4	63.1	62.4	64.9
Concat-Att-2L	58.6	64.8	71.0	66.5	63.5
Concat-Att-4L	60.2	66.9	70.8	64.7	64.9
Cross-Att-4L	60.0	67.2	68.9	68.4	65.0
Concat-Att-6L	59.1	66.4	70.7	65.5	64.4
Cross-Att-6L	52.0	57.6	60.4	55.9	55.4
IPRM(m1,t1)	57.8	65.1	71.0	65.3	63.2
IPRM(m1,t9)	63.1	70.3	76.5	68.9	68.1
IPRM(m6,t1)	62.0	70.2	72.7	68.5	67.5
IPRM(m6,t9)(16)	62.9	70.0	**76.9**	67.1	68.1
IPRM(m6,t9)	**64.2**	**72.9**	75.3	**69.1**	**69.9**

14

B.2 Further comparisons on CLEVR-Humans, CLEVR-CoGenT and NLVRv1

Here, we provide further comparisons with benchmark-specific methods for CLEVR-Humans [33], CLEVR-CoGenT [32] and NLVRv1 [58] (not reported in main paper due to space limitations). As mentioned in main paper, these benchmarks utilize synthetic images and are a test of pure visual reasoning capabilities that are minimally influenced by increased world knowledge or usage of stronger visual backbones.

CLEVR-Humans as already mentioned in main paper evaluates a model's reasoning generalization capabilities to unseen scenarios or question forms. CLEVR-CoGenT studies compositional attribute generalization. Specifically, it has two conditions – i) cond.A wherein all cubes have color $\in \{gray, blue, brown, yellow\}$ and cylinders $\in \{red, green, purple, cyan\}$ (spheres can be any color), and ii) cond.B wherein color-sets are switched b/w cubes and cylinders. A model is then trained on one condition and evaluated on both the original and alternate condition. A higher accuracy on the alternate condition indicates that the model learns more 'compositionally' as it generalizes better to novel shape-color combinations with less feature/attribute combination overfitting.

Table 6: Elaborated results on CLEVR-Humans (left), CLEVR-CoGenT (middle) and NLVRv1 (right). IPRM achieves state-of-art across the three benchmarks and does not require additional supervision such as bounding boxes or functional programs. * requires func. programs supervision / pre-defined dataset-specific neural modules. ▼ requires object bounding-boxes supervision.

Model	CLV-Hum	
	ZS	FT
PG+EE* [33]	54.0	66.6
NS-VQA▼ * [78]	-	67.8
RAMEN [57]	57.8	-
FiLM [55]	56.6	75.9
GLT [4]	-	75.8
LEFT [20]	-	78.8
MAC [26]	57.4	81.5
MDETR▼ [34]	59.9	81.7
IPRM	**63.8**	**85.5**

Model	CoGenTr-A		CoGenFT-B	
	ValA	ValB	ValA	ValB
NS-VQA▼ * [78]	**99.8**	63.9	-	-
MDETR▼ [34]	**99.8**	76.7	-	-
StackAtt-MLP[76]	80.3	68.7	75.7	75.8
PG + EE* [32]	96.6	73.7	76.1	92.7
Tbd-Net* [50]	98.8	75.4	96.9	96.3
MAC [26]	99.0	78.3	97.2	96.1
FILM [55]	98.3	78.8	81.1	96.9
IPRM	99.1	**80.3**	**98.0**	**98.2**

Model	NLVR1
	Test-U
CNN-RNN [58]	56.3
MAC [26]	59.4
FILM [55]	61.2
NMN* [2]	62.0
N2NMN* [22]	66.0
CNN-BiATT [60]	66.1
IPRM (scratch)	63.8
IPRM-CLV-FT	**73.0**

Finally, NLVRv1 evaluates language-grounded visual reasoning. Each sample of this benchmark comprises a set of three synthetic images and a composite natural language statement about the images which can evaluate to True or False and requires various visual-linguistic reasoning skills.

As shown in table 6, IPRM achieves state-of-art results across the three benchmarks and does not require pre-annotated bounding-boxes or functional programs as additional supervision. For **CLEVR-Humans** (table 6 left), it outperforms larger-scale models such as MDETR and RAMEN in zero-shot performance even though the latter is pre-trained on multiple VQA datasets. It also increases state-of-art in finetuned setting by 3.8%.

For **CLEVR-CogenT** (table 6 centre) , IPRM achieves the highest generalization results amongst methods in both the CoGen-Train A and Finetune B. Specifically, it obtains 80.3% acc. on cond. B (when trained on cond. A), which is 1.5% higher than the previous state-of-art cond.B method FILM and 3.6% higher than MDETR. When further finetuned on cond.B, IPRM generalizes for both cond.A and cond.B achieving 98.0% and 98.2% unlike FILM which overfits to cond.B and thereby has poor performance on cond.A. Further, its performance on cond.A (99.1%) is highest amongst methods that do not utilize bounding box or localization supervision and marginally lower than MDETR and NS-VQA (which utilize bounding-box supervision).

Finally, **for NLVRv1** (table 6 right), IPRM model trained from scratch achieves 63.8% acc. and performs competitively with existing task-specific state-of-art model CNN-BiAtt. When finetuned from its CLEVR checkpoint, we find IPRM achieves 73.0% acc. which is 7% higher than existing visual inputs state-of-art for NLVRv1 and suggests strong reasoning transfer capabilities of IPRM. It further outperforms the N2NMN method which requires pre-defined neural modules to be identified for the dataset.

B.3 CLIP Integration Results

We provide results with additional CLIP [56] backbones including CLIP VIT-L/14, CLIP VIT-B/16 and CLIP VIT-L/14@336px on GQA [27], NLVRv2 [59] and CLEVR-Humans in table 7. We compare with alternate prominent vision-language attention mechanisms including Cross-att and

15

Table 7: **Left:** Comparison of IPRM with prominent vision-language attention mechanisms with CLIP VIT-L/14 backbones on CLEVR-Humans, GQA and NLVRv2 benchmarks ('4L' indicates 4 att layers; 'x' indicates model did not converge). **Right:** Results with other CLIP variants VIT-B and VIT-L@ 336 on GQA and NLVRv2. Refer suppl. sec B.3 for further discussion.

Model (CLIP VIT-L/14 bbone)	+Param	+GFLOPs	GQA TestD	NLVR2 Test	CLV-H	
					ZS	FT
Wt-Proj-Fusion	0.6M	0.1	53.5	60.8	58.5	74.4
Cross-Att (2L)	9.2M	1.5	55.1	62.1	-	-
Concat-Att (2L)	7.2M	4.4	55.3	60.5	-	-
Cross-Att (4L)	17.6M	3.1	57.4	54.4	60.3	80.0
Concat-Att (4L)	13.6M	8.9	58.1	55.9	61.2	81.1
Cross-Att (6L)	26.0M	4.5	56.8	x	60.8	80.4
Concat-Att (6L)	19.7M	13.3	57.4	x	62.0	81.8
IPRM	5.2M	5.9	**59.3**	**65.1**	**64.3**	**84.6**

Model (CLIP VIT-B/16 bbone)	GQA TestD	NLVR2 Test
Wt-Proj-Fusion	51.4	59.9
Cross-Att	54.6	56.6
Concat-Att	56.0	57.4
IPRM	55.9	60.8

Model (CLIP VIT-L/14@336)	GQA TestD	NLVR2 Test
Wt-Proj-Fusion	54.0	61.1
Cross-Att	57.4	58.4
Concat-Att	57.3	59.1
IPRM	59.4	65.4

Concat-att blocks as well as a simple joint projection of vision and language pooled representations (referred as Wt-Proj-Att). As shown in the table, IPRM can enhance performance for the CLIP variants across GQA, NLVRv2 and CLV-Humans in comparison to concat and cross-att blocks. Further, it is more parameter efficient with only 5.5M additional parameters in comparison to 4-layer as well as 2-layer stacks of Cross-Att (9.2M 2-layer, 17.6M 4-layer) and Concat-Att (7.2M 2-layer, 13.6M 4-layer). With regards to computational FLOPs, IPRM consumes 5.9GFLOPs which is marginally higher than Cross-Att 4-layer config (3.1GFLOPs) and lower than Concat-Att 4-layer config (8.9GFLOPs). Note, that the performance benefits of adding further layers of cross- or concat-att blocks are observed to be minimal after 4 layers, and can also depend on the amount of training data available. E.g. Both cross- and concat-att blocks of 2 layers had better performances on NLVRv2 (which has a limited set of training questions relative to GQA and CLEVR) in comparison to 4 layer config.

B.4 Further reasoning computation visualizations

We provide elaborate reasoning computation visualizations of IPRM showing the lang. and vis. attentions across parallel operations and computation steps during *operation formation* and *operation execution* stages. Fig. 8 shows a scenario wherein IPRM correctly utilizes parallel and iterative computations to compute intermediate operations of "find object close to front", "retrieve/compare shape and size", "find applicable objects with both same shape and size". Fig. 9 shows another correct prediction of IPRM, and this time, its intermediate reasoning visualization is useful to determine that the entailed reasoning appears sensible. Fig. 10 shows an incorrect prediction by IPRM and its intermediate reasoning visualizations also suggest that IPRM did not understand the question and thereby did not attend to relevant objects. Finally, Fig. 11 shows a scenario wherein while IPRM produces the correct answer, it's intermediate reasoning appears imprecise which makes the prediction (and underlying reasoning) less reliable. We provide further visualizations with a CLIP VIT-L/14 backbone on GQA samples in the supplemental jupyter notebook output (html format for easier viewing).

C Model implementation and experiment details

We implement IPRM in PyTorch [53] as a generic vision-language module receiving a set of input vision (or scene-representation) tokens and input language (or task-representation) tokens. We provide **Python-style pseudocode of IPRM in figs 12, 13 and 14**. For all experiments, we set the internal dimension of IPRM to 512 and use the same configuration of num. parallel operations (N_{op})=6, num. computation steps (T)=9, reduction ratio (r)=2 and window size (W)=2. We follow benchmark-specific conventions for vision-language backbones that are detailed below in sec. C.1. For CLIP [56], we utilize the official models from Huggingface [70]. All experiments are performed on a single NVIDIA A40 GPU with 46GB memory and averaged over 3 trials with different random seeds wherever possible (including STAR, AGQA, CLEVRER-Humans, CLEVR-Humans and GQA). Unless otherwise specified, the learning rate is initialized to 1e-4 with Adam [36] optimizer and gradient clipping value of 8. The learning-rate is reduced based on validation acc. plateau with

16

reduction factor 0.5, threshold 0.001 and patience 0. Further experiment hyper-parameters and settings are provided below. **Source code for experiments and visualization along with model checkpoints will be released publicly via Github**.

C.1 Benchmark-specific experiment details

CLEVR-Humans. We use the CLEVR-Humans dataset from [33] which comprises images from original CLEVR dataset [32] and human crowdsourced questions. We use a batch size of 216 for training. We use the same language encoder (Distil-Roberta[46] from Huggingface[71]) as in existing state-of-art MDETR [34] and frozen ResNet101 backbone layer 3 spatial features (as in [26, 50, 33]). We perform all ablation experiments with 14x14x1024 visual features. Each ablation model is first pretrained for 10 epochs on the original CLEVR dataset (the initial learning rate for IPRM is 1e-4 and for language encoder is 1e-5) and then finetuned on CLEVR-Humans for 40 epochs with early stopping (learning rate of 1e-4 throughout). As observed in prior work [50], we similarly found in multiple scenarios with occluded objects that visual attention only partially identified such objects. Hence, we simply resampled (bilinear sampling) visual input to obtain 16x16x1024 features and empirically found more complete visual attentions with a corresponding 1.1% improvement in accuracy. The final two best performing model configurations (*Nop=6, T=9, W=2, R=2* and *Nop=6, T=9, W=2, R=1*) from ablations were then pre-trained for 35 epochs on CLEVR and finetuned on CLEVR-Humans. While we found that configuration *Nop=6, T=9, W=2, R=1* obtains highest zero-shot (ZS) acc. of 65.6% and finetuned (FT) acc. of 86.3%, we adopt *Nop=6, T=9, W=2, R=2* (with 63.3% ZS and 85.4% FT acc.) as our optimal model given its lesser parameters and FLOPs.

GQA. We use the GQA compositional real-world image question answering dataset from [27]. Based on prior VQA methods on GQA [27, 23, 43, 31], we utilize pre-extracted bounding-box object proposal features and object label predictions obtained from a pretrained object detector [17, 81]. The bounding box coordinates is normalized to range of 0 to 1 based on the original input image size, and the 4 coordinates are transformed to a distributed representation through a learned nonlinear projection. This representation is concatenated with a learned projection of the predicted object labels (initialized with glove[54] 300dim embeddings) to form the final visual input. We train IPRM for 25 epochs with a batch-size of 192 and same hyperparameters as before. We evaluate the final model on both the test-dev split and the official test evaluation server / hidden test set (`https://eval.ai/featured-challenges/225/evaluation`). Our test eval server submission is anonymous with only submission id and method name used ('#7024-IPRM') and a randomly generated team name ('sn12').

STAR-VideoQA. We use the STAR-VideoQA dataset for situational reasoning on real-world videos from [72]. Based on previous videoQA methods [72, 38, 43] for STAR, we utilize object bounding boxes, labels, human pose and human-object relations across frames (note: we do not use the situation hyper-graphs or functional programs). We first perform experiments with the provided ground truth object bounding boxes, labels and human-object relations as well as provided human pose predictions from Alphapose [14] as reported in main paper. Each of these is projected to a distributed representation through learned non-linear projections to obtain object token-wise representations. A further learnable positional embedding for each frame is added to these representations which are then flattened across frames to form the visual input to IPRM. For the language encoder, we found both a simple Bi-LSTM and Distil-roberta language encoder obtain similar performance, and hence choose the simpler Bi-LSTM as the language model. We evaluate models on both 16 uniformly sampled frames and 32 uniformly sampled frames, and empirically found using 32 frames has ∼0.9% higher performance. Since reported models in [72] use 16 frames, we report on the same setting in main paper. A batch size of 64 was used with learning rate 1e-4 over 30 epochs with early stopping. We evaluated the models on both the validation split and official test evaluation server `https://eval.ai/web/challenges/challenge-page/1325/overview`. Our submission is anonymous with only submission id and method name used ('*#9644-IPRM*') and a randomly generated team name ('*sn12*'). For the all-predicted (no ground-truth) visual input setup, similar to [72], we utilize a fasterRCNN [17] object extractor, ST-Trans scene graph extractor [8] and the same Alphapose predictor to obtain predicted object bounding boxes, labels, human-object relations and human poses. We observe a drop of ∼11% in predicted setup similar to observations in [72], suggesting further performance can be achieved through better object and relationship detection backbones.

17

AGQAv2. We use the AGQAv2 [18, 19] benchmark that comprises balanced training and test splits. We followed the same methodology as in STAR. however, since AGQAv2 comprises a larger number of questions and increased diversity in language, we used a distil-roberta language encoder instead of a bi-LSTM.

CLEVRER-Humans. We use the CLEVRER-Humans dataset introduced in [49] for temporal, physical and causal video reasoning which comprises videos from the original CLEVRER dataset [77]. Similar to in STAR-VideoQA and neurosymbolic models [77, 78], we utilize a pretrained faster-RCNN based object localization and attribute prediction network from [77]. We again form object-level representations by concatenating learned projections of object-bounding box coordinates and predicted object attributes (i.e. color, shape and material). A frame-level learnable positional embedding is added and object-tokens across frames are flattened to form the final visual input to IPRM. For the language encoder, we used a simple bi-LSTM similar to existing methods. Note we do not use the functional programs or event causal graphs in our model. The batch size was 128 with a learning rate of 8e-5 and every 4 frames sampled (resulting on average 32 sampled frames). We evaluated models in the three setups – from scratch, zero-shot (CLEVRER-pretrained) and finetuned (CLEVRER-pretrained). Since the CLEVRER-Humans dataset is relatively small (comprising only 1076 questions $\sim$ 8 batches; $\sim$ 0.5% of original CLEVRER), for scratch training we trained for 250 epochs (with early stopping) while for finetuning we finetuned for 150 epochs (with 35 epochs for original CLEVRER training).

CLEVR-CoGen. We use the CLEVR-CoGen dataset from [32] and follow the same setup as in CLEVR-Humans. We use a simpler bi-LSTM language encoder for experiments since the questions are synthetic program-generated unlike in CLEVR-Humans (crowdsourced free-form). We trained our model on condition A for 40 epochs (with early stopping) and used the best cond. A validation performance model to evaluate generalization performance on cond.B. For finetuning on cond.B we finetuned the best cond.A model for 20 epochs and used the best cond.B validation performance model to also evaluate on cond.A. All other hyperparameters are the same as mentioned for CLEVR-Humans.

NLVR. We use the NLVRv1 and NLVRv2 datasets from [58, 59]. NLVRv1 comprises 3 synthetic images and a language statement while NLVRv2 comprises 2 real-world images and a lang. statement. For both datasets, the obtained visual tokens for each images was flattened to obtain the final visual input and an image-wise positional embedding was added to indicate image order. For the language encoder, we used a simple Bi-LSTM.

D   Limitations

Our work proposes a new "iterative" and "parallel" reasoning mechanism (IPRM) designed to address complex visual reasoning and question answering (VQA) scenarios. While we studied IPRM through experiments across various VQA benchmarks along with quantitative ablations and qualitative reasoning visualizations, we note some possible limitations of IPRM in this section. Similar to existing VQA and deep-learning methods, IPRM may reflect biases that are present in the training distribution of VQA benchmarks. This may lead it to overfit to certain image inputs or question forms and possibly provide skewed answers in such scenarios. Further, the utilized vision-language backbones in our experiments may also entail visual, language and cultural biases in their original training distribution which may permeate to IPRM upon integration for VQA scenarios. In this regard, we hope the capability to visualize intermediate reasoning of IPRM and diagnose its error cases (as discussed in main paper Sec. 4.4) can serve a useful tool to benefit interpretability in VQA and identify possible reasoning biases that may emerge in the model.

E   Potential Negative Impact

In relation to VQA and deep-learning methods in general, the deployment of IPRM in real-world applications without thorough consideration of dataset or training distribution biases, could inadvertently reinforce existing vision, language and cultural biases present in the data, leading to erroneous outcomes or skewed answers. Further, the deployment of VQA methods such as IPRM in sensitive domains such as healthcare or scene/footage analysis could raise ethical concerns, including privacy violations, algorithmic reliability, and the potential for unintended consequences stemming from erroneous or biased predictions.

18

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist",**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Abstract and introduction reflects the motivation of method and experimental results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

19

Justification: Limitations are provided in appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Benchmark and training details are provided along with module pseudocode and example CLIP integration. Source code for experiments will made publicly available along with checkpoints.

Guidelines:

20

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Paper provides training and implementations details and provides module pseudocode. Full source code will be made publicly available upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

21

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Appendix provides details on hyperparameters and dataset specific settings for all experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Appendix mentions that results were averaged over atleast 3 seeds for primary experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix mentions the type of GPU and its memory for all experiments along with batch size of experiments.

22

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: Authors reviewed code of ethics during submission.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer:[Yes]

Justification: Paper mentions potential negative impacts of work in appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

23

Answer: answerNA

Justification: Paper does not explicitly pose such risks; however in limitations and potential negative impact this is mentioned.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Yes, all coding libraries and datasets are properly cited and credited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification:No new asset

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

24

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: No new asset

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

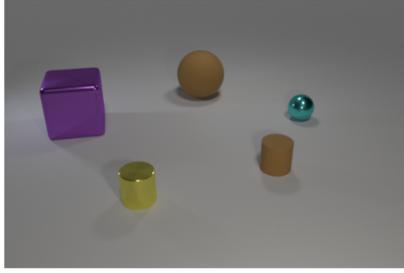
Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No new asset

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

25

Are there any objects that have both shape and size in common with the object that is found closest to the front? (GT: Yes; Pred: Yes)

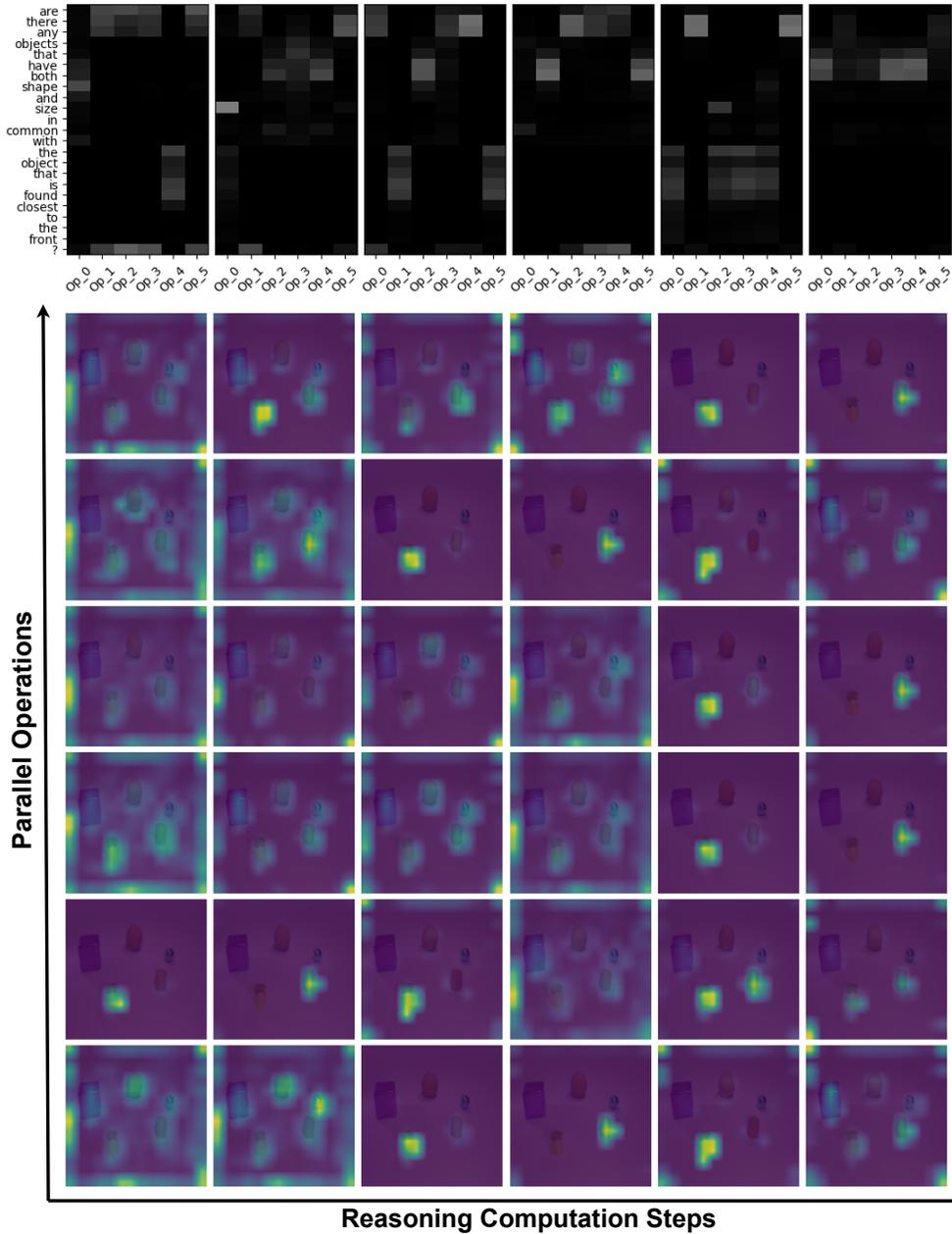
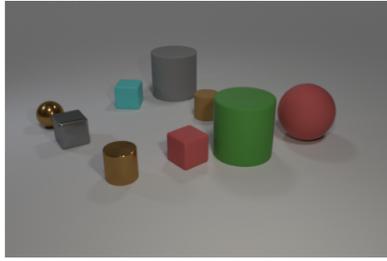



Figure 8: **Top**: original image and question; **middle**: language attentions across parallel operations (clubbed together; op_k represents parallel operation k) and computation steps. **Bottom**: Visual attentions across parallel ops and computation steps. Here, IPRM correctly utilizes parallel and iterative compute to locate the correct candidate object for prediction (to which all operations attend in last step).

26

What shape is the object closest to the gray object with the maximum occuring shape?
Pred: cube **GT**: cube

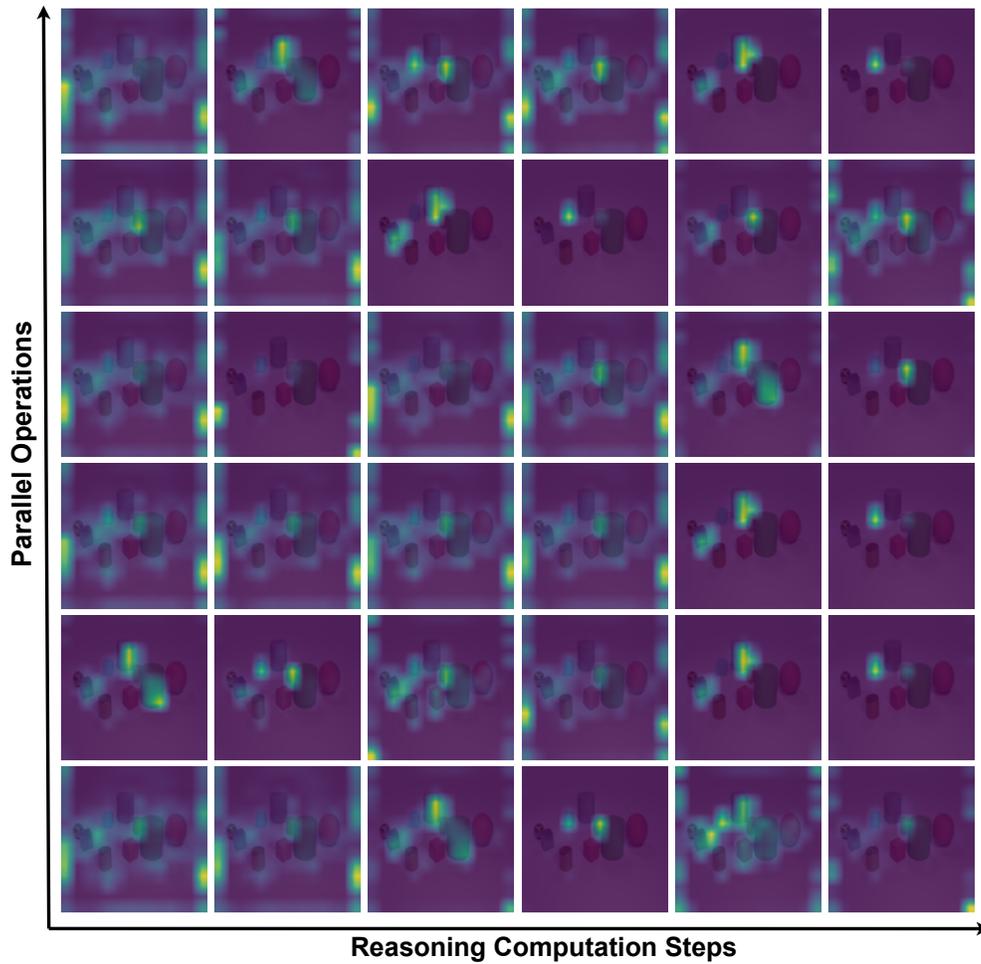
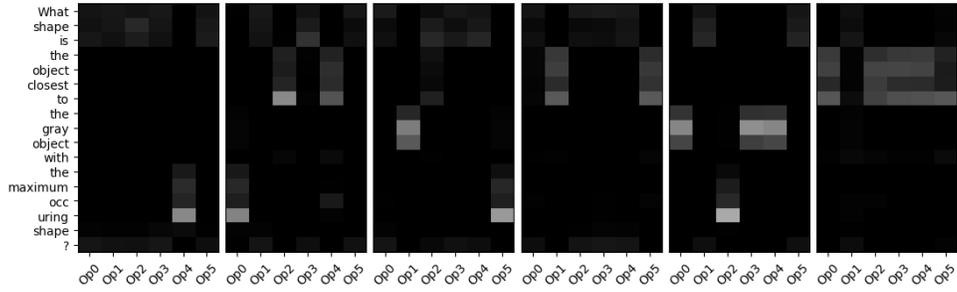
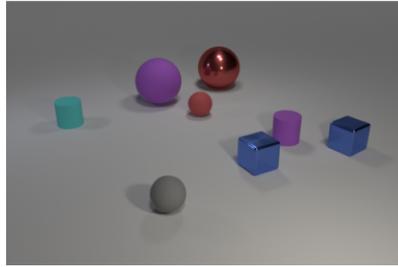



Figure 9: In this example, IPRM predicts the correct answer and its visual attention trace provides evidence of correct intermediate reasoning. In penultimate reasoning step, IPRM correctly localizes the gray object with maximum occuring shape (cylinder) and in the final step, the parallel operations attend to both the cyan cube and the brown cylinder closest to previously identified gray cylinder.

27

Are the two objects that are of a primary color, but not red, of the same shape?
Pred: no GT: yes

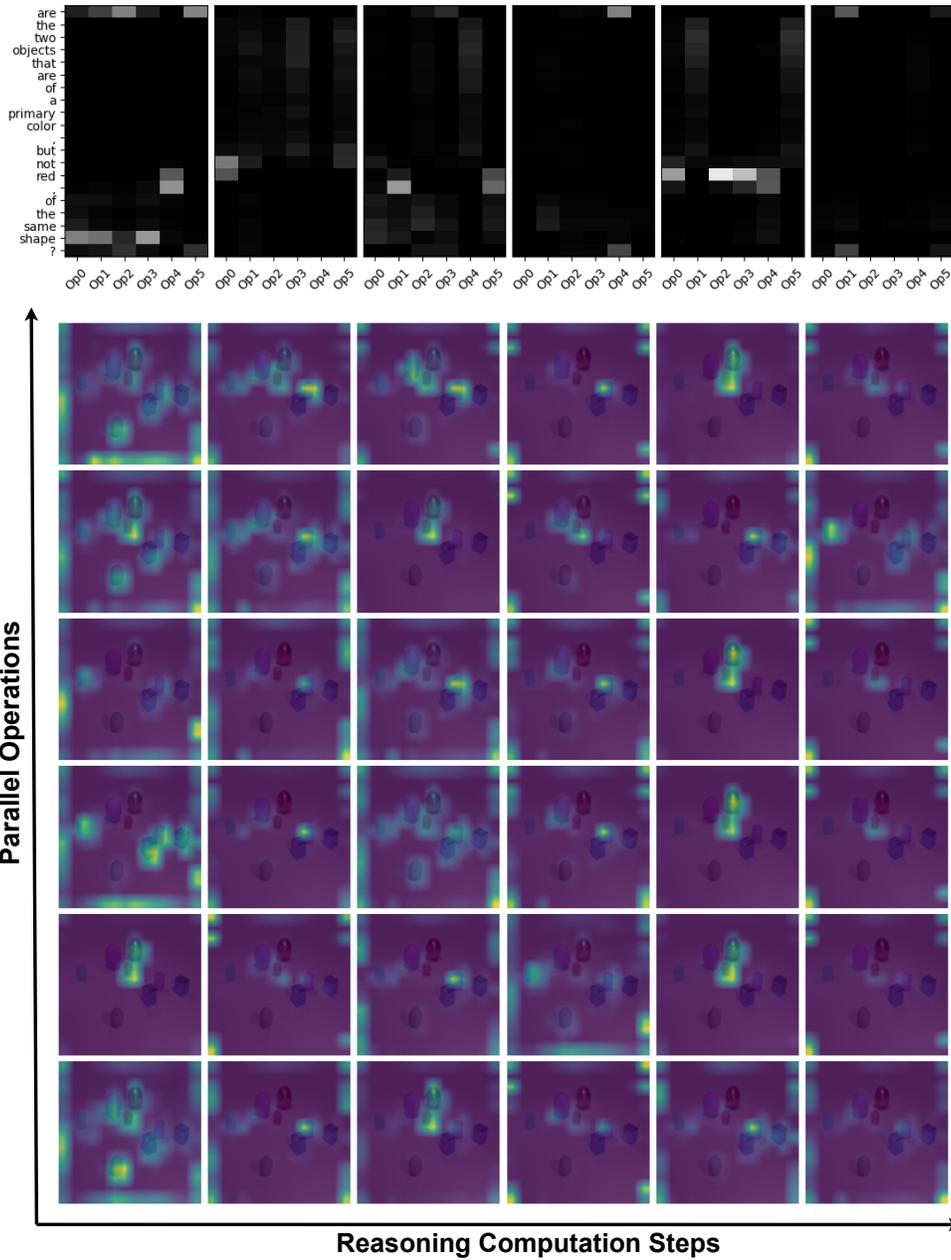
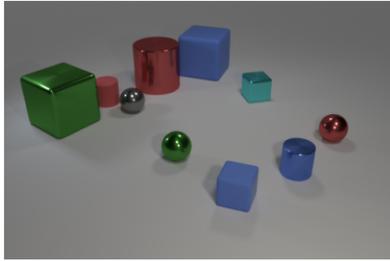



Figure 10: Example where IPRM outputs incorrect answer and the intermediate reasoning appears faulty possibly due to lack of understanding what a "primary color is". The pair of blue (a primary color) cubes in this case should have been identified but are not visually attended in any of the operations across reasoning steps).

28

What shape is the object left of the blue small object with the maximum occuring shape?
Pred: sphere **GT**: sphere

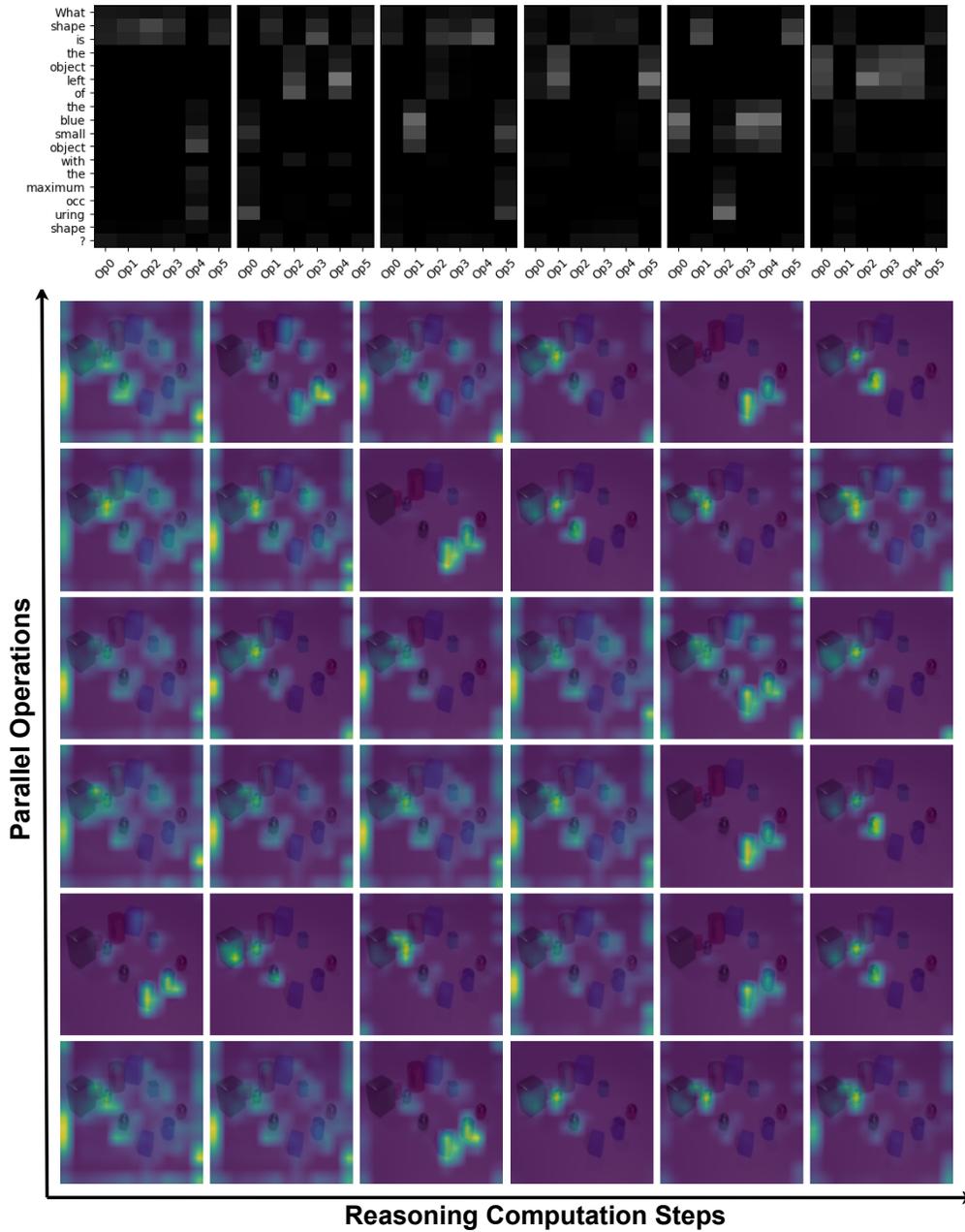



Figure 11: Example wherein IPRM produces correct answer but its visual attention trace suggests intermediate reasoning may be imprecise. The maximum occuring shape is cube; however both the blue small cylinder and blue small cube appear to be attended in the penultimate step as the "blue small object with max occuring shape" making the reasoning and prediction less reliable.

29

```python
def iprm_forward(vis_tokens, #B×Nv×Dm
                 lang_tokens, #B×Nl×Dm
                 lang_summary_rep, #B×Dm
                 num_parallel_ops=6,
                 num_iterative_steps=9,
                 mem_window_len=2):
    mem_op_states = []
    mem_res_states = []
    lang_atts = []
    vis_atts = []

    #0. Initialize memory
    b, d = vis_tokens.size(0), vis_tokens.size(-1)
    mem_op_state, mem_res_state = _init_mem_state(num_parallel_ops, b,d)
    mem_op_states.append(mem_op_state)
    mem_res_states.append(mem_res_state)
    for i in range(num_iterative_steps):
        #1. Form new set of latent operations from lang. token features
        new_ops, lang_att = operation_formation(lang_tokens, mem_op_state)

        #2. Execute new operations on vis. input to form new results
        new_ops_results, vis_att = operation_execution(vis_tokens, new_ops,
        ↪  mem_res_state)

        #3. Apply operation composition
        mem_op_state, mem_res_state = operation_composition(new_ops,
        ↪  new_ops_results, mem_op_states, mem_res_states)

        #4. Maintain memory states within lookback window
        mem_op_states.append(mem_op_state)
        mem_res_states.append(mem_res_state)
        mem_op_states = mem_op_states[min(-1, -mem_window_len):]
        mem_res_states = mem_res_states[min(-1, -mem_window_len):]

        #5. Store lang. and vis. atts for visualization
        lang_atts.append(lang_att)
        vis_atts.append(vis_att)

    #6. 'Pool' final result
    final_result = pool_final_result(mem_res_state, mem_op_state,
    ↪  lang_summary_rep)

    return final_result, lang_atts, vis_atts
```

Figure 12: IPRM pseudocode (1/3)

30

```
1   #Below, "Lin" refers to a linear layer
2   #and `"MLP" refers to a 2-layer multi-layer-perceptron layer
3   def operation_formation(lang_tokens, #B×Nl×Dm
4                            prev_op_state #B×Nop×Dm (Nop=num parallel ops)
5                            ):
6       #1. Form new op "query" based on prior op state
7       op_q = MLP_l(prev_op_state) #paper eq. 4
8
9       #2. Use lang_token_feats as attn "key" and "value" (paper eq. 5)
10      lang_k = lang_tokens
11      lang_v = lang_tokens
12
13      #3. Retrieve new latent ops from lang. rep through attention
14      latent_ops, lang_attn = mod_attn(op_q, lang_k, lang_v,
15                                        lang_attn_proj) #paper eq.6; L194
16
17      return latent_ops, lang_attn
18
19  def operation_execution(vis_tokens, #B×Nv×Dm
20                           new_ops, #B×Nop×Dm
21                           prev_res_state): #B×Nop×Dm
22      #1. Form feature modulation weights (paper eq.7)
23      s_v = concat([Lin_op(new_ops), Lin_res(prev_res_state)]) #concat across feat.
        ↪ axis
24      s_v = Lin_s(s_v)
25
26      #2. Form visual attention "key" (paper eqs. 8 and 9)
27      vis_red_rep = Lin_v1(vis_tokens)
28      mod_vis = s_v * vis_red_rep
29      Nop = mod_vis.size(1)
30      vis_k = MLP_v(concat([mod_vis, vis_red_rep])) #concat across feat. axis
31
32      #3. Form visual attention "query" and "value" (paper eq. 10)
33      vis_q = Lin_op_q(new_ops)
34      vis_v = Lin_v2(vis_tokens)
35
36      #4. Obtain new latent "results" through vis attention (paper eq.11)
37      latent_results, vis_attn = mod_attn(vis_q, vis_k, vis_v, vis_att_proj)
38
39      return latent_results, vis_attn
40
41  def mod_attn(q, k, v, att_proj_layer, attn_mask):
42      qk_mult = q*k  #element-wise product
43      attn_wt = att_proj_layer(qk_mult) #linear projection (paper L194)
44      attn_wt = softmax(attn_wt + (attn_mask * -1e30))
45      out = (attn_wt * v).sum() #sum across feature axis
46      return out, attn_wt
```

Figure 13: IPRM pseudocode (2/3)

31

```python
def operation_composition(new_ops, #B×Nop×Dm
                          new_res, #B×Nop×Dm
                          mem_op_states, #list of W elements: B×Nop×Dm
                          mem_res_states #list of W elements: B×Nop×Dm
                          ):
    #1. Integrate new-ops and results into memory (paper eq. 12 and 13)
    inter_op_state = Lin_op_u(new_ops) + Lin_op_h(mem_op_states[-1])
    inter_res_state= Lin_res_u(new_res) + Lin_res_h(mem_res_states[-1])

    #2. Concat operation and result states over memory lookback window
    op_states_windowed = concat([inter_op_state, mem_op_states])
    res_states_windowed = concat([inter_res_state, mem_res_states])

    #3. Form inter-operation queries and keys (paper eq. 14)
    op_queries = Lin_op_q(inter_op_state)
    op_keys = Lin_op_k(op_states_windowed)

    #4. Form inter-operation op values and res values (paper eq. 15-16)
    op_values = Lin_op_v(op_states_windowed)
    res_values = Lin_res_v(res_states_windowed)

    #5. Compute inter-operation attention (paper eq. 17)
    attn_mask = identity_matrix(op_keys.size(1))[:op_queries.size(1)]
    new_op_state, op_attn_wt = mod_attn(op_queries, op_keys, op_values,
    ↪   op_attn_proj, attn_mask)

    #6. Obtain new operation and result states (paper eq. 18-19)
    new_op_state = new_op_state + Lin_op_u2(inter_op_state)
    new_res_state = op_attn_wt*new_res_state + Lin_res_v2(inter_res_state)

    return new_op_state, new_res_state

def _init_mem_state(num_parallel_ops, b):
    #slice specified num parallel ops from initialized params ~ 𝒩(0,1)
    op_init_state = op_init_param[:num_parallel_ops]
    res_init_state= res_init_param[:num_parallel_ops]
    #broadcast batch-wise to get B×N_op×Dm
    return op_init_state.repeat(b,1,1), res_init_state.repeat(b,1,1)

def pool_final_result(res_state, op_state, lang_summary_rep):
    pool_q = Lin_pq(lang_summary_rep)
    pool_k = Lin_pk(op_state)
    return mod_attn(pool_q, pool_k, res_state)
```

Figure 14: IPRM pseudocode (3/3)

32