
Towards Unsupervised Visual Reasoning: Do Off-The-Shelf Features Know How to Reason?

Monika Wysoczańska^{1*}, Tom Monnier², Tomasz Trzciński^{1,3,4,5}, David Picard²

¹Warsaw University of Technology, ²LIGM, Ecole des Ponts, Univ Gustave Eiffel

³Tooploox ⁴IDEAS NCBR ⁵Jagiellonian University of Cracow

Abstract

Recent advances in visual representation learning allowed to build an abundance of powerful off-the-shelf features that are ready-to-use for numerous downstream tasks. This work aims to assess how well these features preserve information about the objects, such as their spatial location, their visual properties and their relative relationships. We propose to do so by evaluating them in the context of visual reasoning, where multiple objects with complex relationships and different attributes are at play. More specifically, we introduce a protocol to evaluate visual representations for the task of Visual Question Answering. In order to decouple visual feature extraction from reasoning, we design a specific attention-based reasoning module which is trained on the frozen visual representations to be evaluated, in a spirit similar to standard feature evaluations relying on shallow networks. We compare two types of visual representations, densely extracted local features and object-centric ones, against the performances of a perfect image representation using ground truth. Our main findings are two-fold. First, despite excellent performances on classical proxy tasks, such representations fall short for solving complex reasoning problem. Second, object-centric features better preserve the critical information necessary to perform visual reasoning. In our proposed framework we show how to methodologically approach this evaluation.

1 Introduction

Visual representation learning has gained a lot of attention thanks to advanced frameworks demonstrating unprecedented results without relying on explicit supervision. On the one hand, classical self-supervised methods [1, 2, 3, 4, 5] producing localized features (i.e., features that densely correspond to regions of the image) are exhaustively evaluated on standard tasks like image classification or object detection where they perform on par with the supervised ones. On the other hand, more recent unsupervised systems [6, 7, 8, 9] aiming at learning object-centric representations (i.e., each feature is associated with an object in the image) are typically evaluated for instance segmentation where benchmarks are saturated.

In this work, we investigate how well off-the-shelf features extract meaningful information about the objects in a given image. We propose to evaluate the features' ability to model objects through the performance of a reasoning module trained for Visual Question Answering (VQA) by introducing a new VQA evaluation protocol which is based on a simple attention-based reasoning module learned on top of the *frozen* visual features to be evaluated. Similarly to feature evaluations that use shallow networks for predictions, we aim to decouple visual extraction from reasoning. For a fair comparison, we limit the size of the visual features to a small and fixed number. Additionally, we investigate the effect of the training set size to discover what is the minimum size required to be able to learn reasoning patterns and if some types of visual representations are better than others at preventing learning spurious correlations.

*Corresponding author: monika.wysoczanska.dokt@pw.edu.pl

Our VQA evaluation enables us to compare off-the-shelf features, either densely extracted local features or object-centric representations and to make several findings that are the main contributions of this paper: First, such representations have excellent performances using classical proxy tasks, their VQA performance in our constrained setup is far from the one attained by the ground truth or the state-of-the-art obtained using dedicated architectures. With this observation we stress the importance of using complex reasoning tasks, such as VQA, as a complementary evaluation protocol for testing the effectiveness of off-the-shelf image representations. Second, we find that object-centric features are more suited for visual reasoning than local features. Although this is conceptually expected, we provide an empirical way to exhibit this behaviour. Finally, having a limited training set size prohibits learning correct reasoning patterns with all visual representations, although explicit representations seem to be better at preventing learning spurious correlations than implicit ones.

2 Evaluation framework

Our goal is to examine to what extent different off-the-shelf image representations are capable of encoding the information needed for reasoning. We examine popular off-the-shelf features that we split into two groups: (i) the classical dense sets of features localized on a grid-like structure, which we refer to as **LOC**, and (ii) object-centric features which can be associated with objects in the scene, denoted here as **OBJ**. In the case of learned local features, we further distinguish

2 types of learning processes. We examine the standard approach of transferring features obtained from backbones pre-trained for classification on ImageNet [10], which we denote by **IN**. We also study features obtained through more recent self-supervised learning frameworks, denoted by **SSL**.

Pipeline overview We build our pipeline upon the *disentangling reasoning from the vision and language understanding* paradigm [11]. Our framework serves as plug-and-play for unimodal encoders and consists of 3 separate modules, for text, vision and reasoning respectively. First, given a question-image pair we use frozen text and visual encoders to extract features and map them to a common multimodal space. Then, the concatenated features are fed to the reasoning module predicting the answer. An overview of our pipeline is presented in Figure 1.

We use a predetermined and fixed **text encoder** to ensure a fair comparison of the visual features. Precisely, we use the RoBERTa language model introduced in [12], however we note that this could be any other language model that produces a question representation as a sequence of text tokens.

The **visual module** maps an input image to a set of visual tokens V in a two-step process. First, an image encoder extracts an visual representation which is seen as a sequence of feature tokens. Then, to make sure that all features are given the same complexity, we design a module called *memory adapter* whose goal is to convert the extracted representations to a fixed-size input. Inspired by the state-of-the-art model MDETR [13], our **reasoning module** is a transformer encoder-decoder [14] which operates on text and visual token sequences Q and V . To allow our reasoning module to distinguish between modalities, we add to each token a modality-specific segment embedding which is a learnable parameter. In the case of local features, we also add a learned positional encoding to incorporate spatial information, as they do not inherently have such information.

Visual memory adaptation To fix the input memory size of V , we apply a two-step process. First, depending on whether the visual encoder is **OBJ** or **LOC**, the sequence length N_v may differ. Thus, we restrict the N_v to be roughly equal to the maximum number of objects in a scene. Let us assume we have K maximum number of objects that can appear in a scene. Therefore, if $N_v > K$, which is the case for **LOC** due to the grid-like structure of the local features, we apply adaptive average pooling on the features until we roughly match size of K .

Second, to ensure the reasoning module operates on compact representations of similar sizes in memory, we constrain the dimension of the visual tokens. Let us assume d_v^{min} is the minimum visual token dimension needed to solve the task, which corresponds to a number of objects' properties as

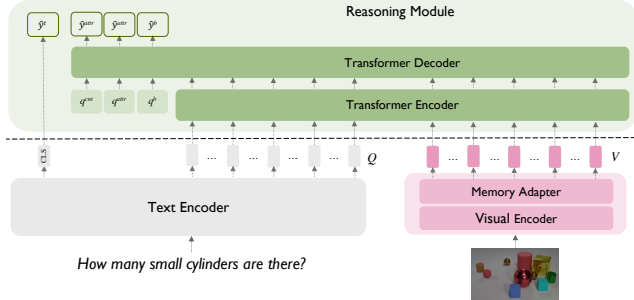


Figure 1: Our evaluation framework overview. During training, only the parameters of the reasoning module are trained (above a dashed line), while the rest of the pipeline remains frozen.

well as their respective positions in a scene for relational reasoning. If $d_v > d_v^{min}$ we compress visual tokens using Principal Component Analysis (PCA) and decrease the dimensionality to match d_v^{min} .

Therefore, we define the minimum memory size M required to solve the task as: $M = K \cdot d_v^{min}$. The memory adapter converts the output of the visual encoder to a fixed memory size within a few orders of magnitude of M by relaxing the d_v^{min} constraint since visual features are not expected to attain perfect compression of the visual information.

Training strategy We train our reasoning module using question-image-answer triplets (q, i, a) without any external supervision. Concretely, given a triplet we predict the answer type \hat{y}_t as well as the corresponding answer encoded by \hat{y}_b , \hat{y}_{cnt} or \hat{y}_{attr} respectively associated to *True/False*, *count* and *attribute* questions. Our final loss is defined as: $L_{total} = L_t + L_b + L_{cnt} + L_{attr}$, where L_t , L_{cnt} , L_{attr} denote cross entropy losses between the ground truth y_t, y_{cnt}, y_{attr} , and the predictions $\hat{y}_t, \hat{y}_{cnt}, \hat{y}_{attr}$, whereas L_b is a binary cross entropy loss between y_b and \hat{y}_b .

3 Experiments

3.1 Implementation details

For our comparative study, we choose popular image representations trained with various levels of supervision. Models are always frozen during training and used solely as feature extractors. We use features extracted from two popular architectures: convolutional-based ResNet-50 [15] and the transformer-based ViT-S proposed in [16]. To study the influence of the level of supervision in image backbones, in the case of ResNet-50 we evaluate the local features trained in a supervised manner using the classification task on Imagenet, denoted **LOC-IN**, and self-supervised DINO features, denoted **LOC-SSL**. For ViT-S, we use features trained with DINO. To evaluate the performance of **OBJ** we consider two methods: Slot Attention [8] and DTI-Sprites [9] which demonstrated state-of-the-art segmentation results on the recent CLEVRText benchmark [17].

We conduct our analysis on the CLEVR dataset [18] - a common synthetic benchmark for Visual Question Answering. CLEVR world consists of 3D objects in different shapes, colours, sizes and materials. The maximum number of objects in scene K is 10. Therefore, the minimum memory size of visual tokens $M = 10 \times 7$, for 10 objects, 4 properties and 3D position (x,y,z).

Since all considered representations are much larger than M , we consider 2 memory sizes in our protocol: (i) a total memory of size 100, called 100 mem size, which approximates the min M for CLEVR dataset, and (ii) a total memory of size 1000, called 1000 mem size, to account for richer representations than the bare minimum required for solving the VQA task.

All the details on adaptation and implementations can be found in Appendix A.1.

3.2 Results and discussion

We conduct 2 experiments. In the memory size constraint test, we study the effectiveness of different visual features in solving the VQA task when heavily constraining the input size of the reasoning module. The overall accuracy as well as all the question-specific categorical accuracy are shown in Table 1. Moreover, we study the effectiveness of visual features in solving VQA with a limited number of samples. The idea behind this experiment is to test the generalization capacity of the features as we argue that accurate and discriminative features should generalize better to unseen samples. Figure 2 shows a comparison for varying fractions of the training set for both memory sizes.

Are visual representations anywhere close to ground truth performances? From the 100 mem size setup it is clear that all of the methods except for DTI-Sprites fail to attain an accuracy which further suggests that they do not encode visual information in a compact way that is comparable to

	method	overall	count	exist	comp num	query attr	comp attr
<i>100 mem size</i>							
	GT	99.7	99.8	99.9	99.5	99.2	100.0
OBJ	DTI-Sprites [9]	81.8	72.6	87.8	84.8	84.3	82.4
OBJ	Slot Attention [8]	58.9	50.8	70.8	70.1	58.3	55.8
LOC-SSL	DINO ResNet-50	60.4	51.0	72.6	71.4	58.7	61.2
LOC-SSL	DINO ViT	52.1	45.2	66.9	69.2	46.3	53.1
LOC-IN	ResNet-50	57.1	47.3	69.3	69.7	56.2	56.0
LOC	Raw	48.9	44.9	64.1	67.9	39.8	51.1
<i>1000 mem size</i>							
	GT	99.8	100.0	100.0	99.8	99.1	100.0
OBJ	Slot Attention [8]	91.4	85.1	95.0	92.8	93.1	92.6
OBJ	DTI-Sprites [9]	82.2	72.7	88.1	86.2	84.6	82.9
LOC-SSL	DINO ResNet-50	82.4	71.4	88.8	82.9	85.7	84.7
LOC-SSL	DINO ViT	78.4	66.0	86.5	79.1	80.7	81.0
LOC-IN	ResNet-50	79.0	66.6	85.9	81.5	83.0	80.2
LOC	Raw	50.5	44.5	64.8	69.0	43.9	51.2
	MDETR [13]	99.7	99.3	99.9	99.4	99.9	99.9

Table 1: Results on the CLEVR dataset. We report scores on the validation set with a detailed split by the question type. Overall, even in the larger memory setup, there is a significant gap between off-the-shelf features and the state of the art MDETR [13], which performs on par with ground truth.

perfect visual information. We attribute the overall best performance of DTI-Sprites in this restrictive memory size regime to its inherent explicit representation nature. The original memory size of a DTI-Sprites visual token is the closest to the ground truth, which plays in its favour in this experiment.

Do visual representations contain sufficient information for solving VQA?

In the more relaxed memory constraint setup (1000 mem size), all the studied visual representations enable the reasoning module to learn to solve VQA to some extent, with Slot Attention clearly outperforming the rest. This may suggest that even if the visual information is not encoded in a way that can be heavily compressed, these features nonetheless contain significant semantic information. To study the effect of introducing memory constraint we also train our reasoning module using original feature sizes. In case of DINO ResNet-50 features we obtained 87.9% overall accuracy, which indicates that our memory adapter does introduce a bottleneck. Nevertheless, features at their best are still far from the ground truth performance.

Are object-centric representations more suitable for the task?

For the *exist* questions, **OBJ** features are outperforming **LOC** ones. This is expected and quantitatively indicates that structuring the scene into a set of objects is much more suitable for reasoning. When it comes to *comp num*, *query attr* and *comp attr*, Slot Attention performs significantly better compared to other methods. We argue this can be attributed to the object-centric nature of the representation that facilitates comparisons among objects and focuses on describing their properties. We note that **OBJ** indeed were trained on CLEVR contrary to **LOC** features. However, we tried fine-tuning DINO features on CLEVR dataset but we obtained worse results. We hypothesize they are not suited for synthetic datasets like CLEVR, which is smaller and much less diverse than standard SSL datasets.

Are visual representations able to reason from a few examples? We observe that starting at 20% of the full training set and with perfect visual information corresponding to the ground truth, it is possible to infer reasoning patterns. However, at this training set size, none of the considered visual representations enables visual reasoning, regardless of the memory constraint.

Are compact and explicit visual representations better suited for learning to reason from a few examples?

Looking at the 100 mem size setup (Figure 2a), we can see that DTI-Sprites is able to obtain higher accuracy with much fewer examples than the other methods, whereas all achieve comparable very low training losses (see A.3). This may indicate that explicit representations enable a quicker discovery of relevant information than implicit representations.

Do object-centric representations prevent learning spurious correlation? Figure 2b suggests that both **OBJ** and **LOC** representations exhibit similar behaviour when the memory is less constrained. Given that they all reach similar very high training accuracy, this indicates that structuring the representation into objects may not be as good at preventing learning spurious correlations as is the distinction between explicit and implicit representations.

4 Conclusions

We investigated to what extent off-the-shelf representations model the information necessary to perform visual reasoning. To that end, we design a new feature evaluation protocol based on VQA which aims at disentangling as much as possible the vision from the reasoning part. Using our evaluation protocol, we make three key findings: (i) off-the-shelf visual representations are far from being able to structure visual information in a compact manner, (ii) object-centric representations seem to be better at encoding the critical information necessary for reasoning, and (iii) limiting the training set size has a dramatic impact on the learning of spurious correlations. While these findings

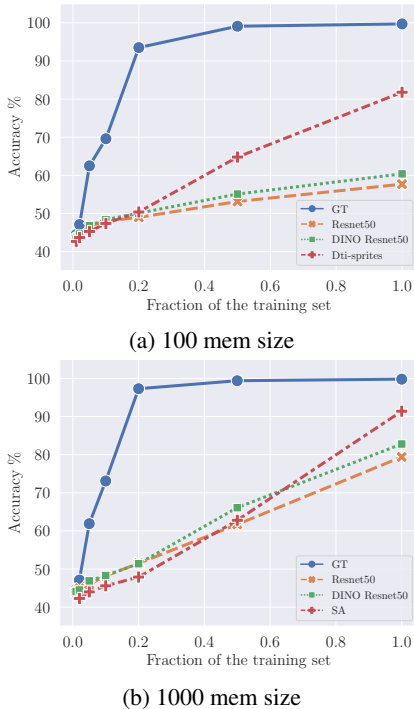


Figure 2: Results of the few-shot VQA experiments on CLEVR dataset.

contrast with the excellent performances that off-the-shelf features usually obtain in simpler vision tasks, they also show that having representations that encode object properties is a promising first step towards unsupervised visual reasoning.

Acknowledgments and Disclosure of Funding

This work was supported by HPC resources from GENCI-IDRIS (2021-AD011013085), by Foundation for Polish Science (grant no POIR.04.04.00-00-14DE/18-00) carried out within the Team-Net program co-financed by the European Union under the European Regional Development Fund, as well as the National Centre of Science (Poland) Grant No. 2020/39/B/ST6/01511.

References

- [1] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. *arXiv preprint arXiv:1805.01978*, 2018.
- [2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [3] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent—a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.
- [4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.
- [5] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.
- [6] Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. MONet: Unsupervised Scene Decomposition and Representation. *arXiv:1901.11390*, 2019.
- [7] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-Object Representation Learning with Iterative Variational Inference. In *ICML*, 2019.
- [8] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *CoRR*, abs/2006.15055, 2020.
- [9] Tom Monnier, Elliot Vincent, Jean Ponce, and Mathieu Aubry. Unsupervised Layered Image Decomposition into Object Prototypes. In *ICCV*, 2021.
- [10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.
- [11] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B. Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. In *NeurIPS*, 2018.
- [12] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.
- [13] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021.

- [14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.
- [17] Laurynas Karazija, Iro Laina, and Christian Rupprecht. Clevrtex: A texture-rich benchmark for unsupervised multi-object segmentation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.
- [18] 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 *CVPR*, 2017.
- [19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [21] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface’s transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.
- [22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

A Appendix

A.1 Visual encoders extraction and adaptation details

A.1.1 Dense local features

We use features extracted from two popular architectures: convolutional-based ResNet-50 [15] and the transformer-based ViT-S proposed in [16]. For ResNet-50, we use the local features after the last convolutional layer right before the global average pooling, whereas for ViT-S, we use the output tokens of the last layer corresponding to the image patch position (*i.e.*, the CLS token is not used). To encode the position of local features, we use the corresponding 2D position in the feature map for ResNet-50 features and the 2D position of the corresponding patch for ViT-S features.

For ResNet-50 pre-trained on ImageNet we use model weights available in torchvision package². For both DINO ResNet-50 and DINO ViT-S/16 we use weights provided in original DINO repository³.

A.1.2 Object-centric representations

To evaluate the performance of unsupervised object-centric representations, we consider two methods: Slot Attention [8] and DTI-Sprites [9] which demonstrated state-of-the-art segmentation results on the recent CLEVRText benchmark [17]. We first train both methods on CLEVR dataset - which does not require any supervision - and use pre-trained models as feature extractors.

Slot Attention In Slot Attention we use the slots right before the decoder part as features. We use the implementation available in CLEVRText benchmark repository⁴. We train the model on the original, CLEVR VQA dataset, preserving the original train/validation splits. To keep a similar resolution as in the case of the multi-object segmentation task, we feed input images resized to 120×160 . We do not apply any centre cropping to make sure all the objects in the scenes are clearly visible. Following [17] we use 11 slots, we also maintain the original learning rate, batch size, and optimizer settings as well as 500k iterations of training.

DTI-Sprites We use original implementation of DTI-Sprites⁵. Similar to Slot Attention, we train the model on images resized to 120×160 with no centre cropping. To account for smaller resolution compared to the original multi-object segmentation task, we increase the representation expressiveness in the backbone by changing adaptive average pooling to 4×4 , instead of the originally proposed 2×2 . We train the model with 10 layers, corresponding to a maximum number of objects in the CLEVR dataset. We also increase the number of prototypes to 8 since we observed that the training did not lead to obtaining a complete set of prototypes in the dataset when using only 6. We train the model with the original batch size and learning rate for 760k iterations.

Memory adaptation details To match the memory constraints in 2 setups for all visual encoders we use PCA implementation in Scikit-learn [19]. We first extract features for train and validation sets and apply PCA offline.

A.2 Training details

We implement our framework in PyTorch [20]. For the text encoder we use RoBERTa⁶ model available in HuggingFace library [21]. In the case of all visual encoders, we follow the same training strategy. We train the reasoning module for 40 epochs, with a batch size of 64. We use adamW [22] optimizer with a learning rate of $1e-4$, weight decay of $1e-4$ and linear warmup for the first 10k iterations. We then decrease the learning rate at epochs 30th and 35th by a factor of 10.

²<http://pytorch.org/vision/stable/index.html>

³<http://github.com/facebookresearch/dino>

⁴<http://github.com/karazijal/clevrtex>

⁵<http://github.com/monniert/dti-sprites>

⁶<http://huggingface.co/roberta-base>

Table 2: Memory size constraints imposed at the feature adaptation step. We conduct our studies in 2 memory size regimes: *100 mem size*, and *1000 mem size*. We also provide the raw output sizes for each visual encoder to highlight the extent of compression and expansion applied at the adaptation stage.

method	input image size	raw output size	100 mem size	1000 mem size
GT	-	10×7	10×10	10×100
DTI-Sprites	120×160	10×10	10×10	10×100
Slot Attention	120×160	11×64	11×9	11×90
ResNet-50	224×224	49×2048	16×6	16×62
ViT-S	224×224	194×384	16×6	16×62
Raw	192×192	$9 \times 12k$	9×11	9×111

Computational cost Regardless of the visual encoder used as an input, a full training run takes approximately 20 hours on a single NVIDIA Tesla V100 GPU. In total we used approximately 15k GPU hours for both obtaining visual features as well as for the evaluation phase.

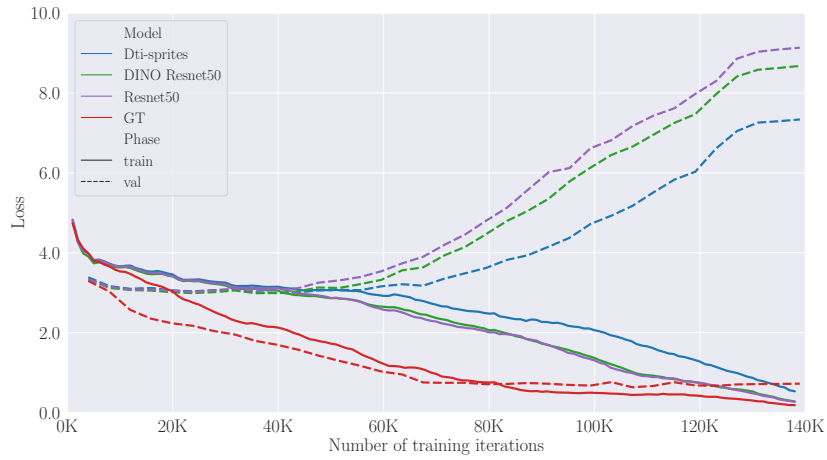
A.3 Details on the training set size study

In the training set study, we limit the number of training samples by reducing the number of scenes. We observe that starting at 20% of the full training set with perfect visual information corresponding to the ground truth, it is possible to obtain high validation accuracy. Therefore, we study the effectiveness of visual encoders at this threshold.

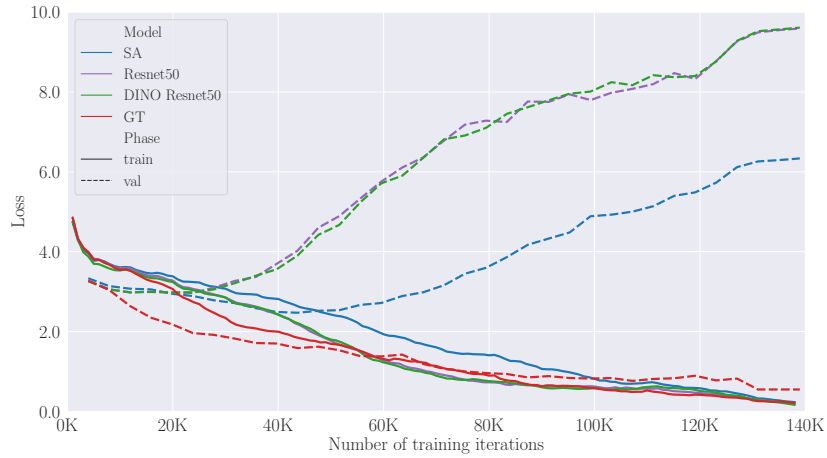
Figure 3 shows the learning curves at a 20% fraction of the training set for both memory size regimes. In Figure 3a, which depicts the learning curves in 100 mem size, in the case of all of the methods we observe that the model starts to quickly overfit, with DTI-Sprites performing relatively better.

In the case of 1000 mem, Figure 3b size we observe the overfitting effect happening even earlier in the training process. This may indicate the occurrence of leakage from visual encoders to the reasoning part.

In both memory size regimes **OBJ** representations demonstrate relatively better generalization capabilities but are still far from the ground truth information.



(a) 100 mem size



(b) 1000 mem size

Figure 3: Training and validation curves at a 20% fraction of the training set. We report training loss for 20% of the training set being used, while validation loss we calculate over the whole validation set in the CLEVR dataset.