BLOCK: Bilinear Superdiagonal Fusion for Visual Question Answering and Visual Relationship Detection

Hedi Ben-Younes, Remi Cadene, Nicolas Thome, Matthieu Cord

BLOCK: Bilinear Superdiagonal Fusion for Visual Question Answering and Visual Relationship Detection

Hedi Ben-younes[1,2] **Remi Cadene**[1] **Nicolas Thome**[3] **Matthieu Cord**[1]
[1]Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
[2]Heuritech
[3]CEDRIC - Conservatoire National des Arts et Métiers, 75003 Paris, France

Abstract

Multimodal representation learning is gaining more and more interest within the deep learning community. While bilinear models provide an interesting framework to find subtle combination of modalities, their number of parameters grows quadratically with the input dimensions, making their practical implementation within classical deep learning pipelines challenging. In this paper, we introduce BLOCK, a new multimodal fusion based on the block-superdiagonal tensor decomposition. It leverages the notion of block-term ranks, which generalizes both concepts of rank and mode ranks for tensors, already used for multimodal fusion. It allows to define new ways for optimizing the tradeoff between the expressiveness and complexity of the fusion model, and is able to represent very fine interactions between modalities while maintaining powerful mono-modal representations. We demonstrate the practical interest of our fusion model by using BLOCK for two challenging tasks: Visual Question Answering (VQA) and Visual Relationship Detection (VRD), where we design end-to-end learnable architectures for representing relevant interactions between modalities. Through extensive experiments, we show that BLOCK compares favorably with respect to state-of-the-art multimodal fusion models for both VQA and VRD tasks. Our code is available at `https://github.com/Cadene/block.bootstrap.pytorch`.

1 Introduction

In many of the most recent tasks tackled by artificial intelligence, multiple sources of information need to be taken into account for decision making. Problems such as visual question answering (Goyal et al. 2017), visual relationship detection (Lu et al. 2016), cross-modal retrieval (Kiros, Salakhutdinov, and Zemel 2015) or social-media post classification (Duong, Lebret, and Aberer 2017) require, at a certain level and to some extent, the fusion between multiple modalities.

For classical mono-modal tasks, linear models constitute a handful building block to transform the input $\boldsymbol{x}$ and to match it with the desired output $\boldsymbol{y}$. When dealing with two modalities, not only do we need to properly transform each input $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ into a representation that fits the problem, we also want to model the interactions between these modalities. A natural candidate to extend linear models for two inputs are bilinear models. However, the number of parameters of a bilinear model is quadratic in the input dimensions: as a linear model is characterized by a matrix $\boldsymbol{W} \in \mathbb{R}^{dim(\boldsymbol{x}) \times dim(\boldsymbol{y})}$, a bilinear model is defined by a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{dim(\boldsymbol{x}_1) \times dim(\boldsymbol{x}_2) \times dim(\boldsymbol{y})}$. When the input dimensions grow (thousand or more), learning a full tensor $\boldsymbol{\mathcal{T}}$ becomes quickly intractable. The main issue is to reduce and control the numbers of parameters representing $\boldsymbol{\mathcal{T}}$. Usually, when working with linear models, restricting the complexity is done by constraining the rank of the matrix $\boldsymbol{W}$. Unfortunately, it is much more complicated when it comes to bilinear models, since it requires notions of multi-linear algebra. While a lot of work has been done in extending the notion of complexity to higher-order tensors (Harshman et al. 2001), (Carroll and Chang 1970), (Tucker 1966), it is not clear whether the simple extension of *rank* should be used to constrain a bilinear model.

In this paper, we tackle the general problem of learning end-to-end bilinear models. We propose BLOCK, a Block Superdiagonal Fusion framework for multimodal representation based on the block-term tensor decomposition (De Lathauwer 2008). As it has been studied in the signal processing literature (Cichocki et al. 2015), the focus was on having uniqueness properties to ensure a physically interpretable decomposition. We study here this decomposition under a machine learning perspective instead, and use it as a fully learnable tensor of parameters. Interestingly, this decomposition leverages the notion of block-term ranks to define a tensor's complexity. It encapsulates both concepts of *rank* and *mode ranks*, at the basis of Candecomp/PARAFAC (Harshman et al. 2001) and Tucker decomposition (Tucker 1966). This complexity analysis is capitalized on to provides a new way to control the tradeoff between the expressiveness and complexity of the fusion model. More precisely, BLOCK enables to model very rich (*i.e.* full bilinear) interactions between groups of features, while the block structure limits the whole complexity of the model, which enables to keep expressive (*i.e.* high dimensional) mono-moodal representations.

The BLOCK model is used for solving two challenging applications: Visual Question Answering (VQA) and Visual Relationship Detection (VRD). For both tasks, the number of blocks and the size of each projection in the BLOCK fusion will be adapted to balance between fine interaction

modeling and low number of parameters. We embed our bilinear BLOCK fusion strategy into deep learning architectures ; through extensive experiments, we validate the relevance of the approach as we provide an extensive and systemic comparison of many state-of-the-art multimodal fusion techniques. Moreover, we obtain very competitive results on three commonly used datasets: VQA 2.0 (Goyal et al. 2017), TDIUC (Kafle and Kanan 2017) and the VRD dataset (Lu et al. 2016).

2 BLOCK fusion model

In this section, we present our BLOCK fusion strategy and discuss its connection to other bilinear fusion methods from the literature.

A bilinear model takes as input two vectors $\boldsymbol{x}^1 \in \mathbb{R}^I$ and $\boldsymbol{x}^2 \in \mathbb{R}^J$, and projects them to a K-dimensional space with tensor products:

$$\boldsymbol{y} = \boldsymbol{\mathcal{T}} \times_1 \boldsymbol{x}^1 \times_2 \boldsymbol{x}^2 \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^K$. Each component of $\boldsymbol{y}$ is a quadratic form of the inputs: $\forall k \in [1, K]$,

$$y_k = \sum_{i=1}^{I} \sum_{j=1}^{J} \boldsymbol{\mathcal{T}}_{ijk} . x_i^1 . x_j^2 \tag{2}$$

A bilinear model is completely defined by its associated tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I \times J \times K}$, the same way as a linear model is defined by its associated matrix.

2.1 BLOCK model

In order to reduce the number of parameters and constrain the model's complexity, we express $\boldsymbol{\mathcal{T}}$ using the block-term decomposition. More precisely, the decomposition of $\boldsymbol{\mathcal{T}}$ in rank (L,M,N) terms is defined as:

$$\boldsymbol{\mathcal{T}} := \sum_{r=1}^{R} \boldsymbol{\mathcal{D}}_r \times_1 \boldsymbol{A}_r \times_2 \boldsymbol{B}_r \times_3 \boldsymbol{C}_r \tag{3}$$

where $\forall r \in [1, R]$, $\boldsymbol{\mathcal{D}}_r \in \mathbb{R}^{L \times M \times N}$, $\boldsymbol{A}_r \in \mathbb{R}^{I \times L}$, $\boldsymbol{B}_r \in \mathbb{R}^{J \times M}$ and $\boldsymbol{C}_r \in \mathbb{R}^{K \times N}$. This decomposition is called *block-term* because it can be written as

$$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{D}}^{bd} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C} \tag{4}$$

where $\boldsymbol{A} = [\boldsymbol{A}_1, ..., \boldsymbol{A}_R]$ (same for $\boldsymbol{B}$ and $\boldsymbol{C}$), and $\boldsymbol{\mathcal{D}}^{bd} \in \mathbb{R}^{LR \times MR \times NR}$ the block-superdiagonal tensor of $\{\boldsymbol{\mathcal{D}}_r\}_{1 \leq r \leq R}$, as illustrated in Figure 1. Applying this structural constraint to $\boldsymbol{\mathcal{T}}$ in Eq. (1), we can express $\boldsymbol{y}$ with respect to $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$. Let $\hat{\boldsymbol{x}}^1 = \boldsymbol{A}\boldsymbol{x}^1 \in \mathbb{R}^{LR}$ and $\hat{\boldsymbol{x}}^2 = \boldsymbol{B}\boldsymbol{x}^2 \in \mathbb{R}^{MR}$. These two projections are merged with a fusion parametrized by the block-superdiagonal tensor $\boldsymbol{\mathcal{D}}^{bd}$. Each block in this tensor merges together chunks of size L from $\hat{\boldsymbol{x}}^1$ and of size M from $\hat{\boldsymbol{x}}^2$ to produce a vector of size N:

$$\boldsymbol{z}_r = \boldsymbol{\mathcal{D}}_r \times_1 \hat{\boldsymbol{x}}^1_{rL:(r+1)L} \times_2 \hat{\boldsymbol{x}}^2_{rM:(r+1)M} \tag{5}$$

where $\hat{\boldsymbol{x}}_{i:j}$ is a vector of dimension $j - i$ containing the corresponding values in $\hat{\boldsymbol{x}}$. Finally, all the $\boldsymbol{z}_r$ vectors are concatenated to produce $\boldsymbol{z} \in \mathbb{R}^{NR}$. The final prediction vector is $\boldsymbol{y} = \boldsymbol{C}\boldsymbol{z} \in \mathbb{R}^K$.

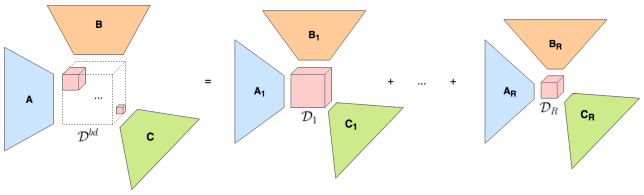


Figure 1: BLOCK framework. The third-order interaction tensor is decomposed in R rank-(L,M,N) terms. We give here the two equivalent representations of the block-term decomposition, presented in this paper. On the left, we show the formulation with the block-superdiagonal tensor decomposition that corresponds to Eq. (4). On the right, we express it as a sum of small decompositions, as written in Eq. (3). Through the R number of blocks and the dimensions of the $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}$ projections, we can handle the trade-off between model complexity and expressivity.

To further reduce the number of parameters in the model, we add a constraint on the rank of each third order slices matrices of the blocks $\boldsymbol{\mathcal{D}}_r$, as it was done in some recent VQA applications (see Section 3).

Discussion When working with a linear model, a usual technique to restrict the hypothesis space and number of parameters is to constrain the rank of its associated matrix. The formal notion of *matrix rank* quantifies how complex a linear model is allowed to be. However, when it comes to restricting the complexity of a bilinear model, multiple algebraic concepts can be used. We give two examples that are related to the block-term decomposition.

The Candecomp/PARAFAC (CP) decomposition (Carroll and Chang 1970), (Harshman et al. 2001) of a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I \times J \times K}$ is the linear combination of rank-1 terms

$$\boldsymbol{\mathcal{T}} := \sum_{r=1}^{R} \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r \tag{6}$$

where $\circ$ denotes the outer product, and the vectors $\boldsymbol{a}_r \in \mathbb{R}^I$, $\boldsymbol{b}_r \in \mathbb{R}^J$ and $\boldsymbol{c}_r \in \mathbb{R}^K$ represent the elements of the decomposition. The rank of $\boldsymbol{\mathcal{T}}$ is defined by the minimal number R of triplet vectors so that the Eq. (6) is true. Thus, restricting the hypothesis space for $\boldsymbol{\mathcal{T}}$ to the set of tensors defined by Eq. (6) guarantees that the rank is upper-bounded by R.

Applying this constraint on $\boldsymbol{\mathcal{T}}$, Eq. (1) is simplified into

$$\boldsymbol{y} = \boldsymbol{C}\left(\left(\boldsymbol{x}^{1\top}\boldsymbol{A}\right) * \left(\boldsymbol{x}^{2\top}\boldsymbol{B}\right)\right) \tag{7}$$

where $\boldsymbol{A} = [\boldsymbol{a}_1, ..., \boldsymbol{a}_R] \in \mathbb{R}^{I \times R}$, $\boldsymbol{B} = [\boldsymbol{b}_1, ..., \boldsymbol{b}_R] \in \mathbb{R}^{J \times R}$ and $\boldsymbol{C} = [\boldsymbol{c}_1, ..., \boldsymbol{c}_R] \in \mathbb{R}^{K \times R}$, and $*$ denotes element-wise product. This decompositioncan be seen as a special case of the block-term decomposition where $L = M = N = 1$, reducing $\boldsymbol{\mathcal{D}}^{bd}$ to a super-diagonal identity tensor.

Another way to restrict a three-way tensor's complexity is through its *mode ranks*. The rank-(L,M,N) Tucker decomposition (Tucker 1966) of $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I \times J \times K}$ is defined as

$$\boldsymbol{\mathcal{T}} := \boldsymbol{\mathcal{D}} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C} \tag{8}$$

where $\mathcal{D} \in \mathbb{R}^{L \times M \times N}$, $\boldsymbol{A} \in \mathbb{R}^{I \times L}$, $\boldsymbol{B} \in \mathbb{R}^{J \times M}$ and $\boldsymbol{C} \in \mathbb{R}^{K \times N}$. This decomposition assumes a constraint on the three unfolding matrices of $\mathcal{T}$, such that $\text{Rank}\left(\mathcal{T}_{JK \times I}\right) = L$, $\text{Rank}\left(\mathcal{T}_{KI \times J}\right) = M$ and $\text{Rank}\left(\mathcal{T}_{IJ \times K}\right) = N$ (following the notations in (De Lathauwer 2008)).

Applying this constraint to $\mathcal{T}$, Eq. (1) can be re-written as:

$$\boldsymbol{y} = \boldsymbol{C}\left(\mathcal{D} \times_1 \left(\boldsymbol{x}^{1\top}\boldsymbol{A}\right) \times_2 \left(\boldsymbol{x}^{2\top}\boldsymbol{B}\right)\right) \quad (9)$$

This decomposition can be seen as a special case of the block-term decomposition where there is only $R = 1$ block in the core tensor.

As was studied by (De Lathauwer 2008), the notion of tensor complexity should be expressed not only in terms of rank or mode ranks, but using the number of blocks and the mode-n ranks of each block. It appears that CP and Tucker decompositions are two extreme cases, where only one of the two quantities is used. For the CP, the number of blocks corresponds to the rank, but each block is of size (1,1,1). The monomodal projections can be high-dimensional and thus integrate rich transformation of the input, but the interactions between both projections is relatively poor as a dimension from one space is only allowed to interact with another. For the Tucker decomposition, there is only one block of size (L,M,N). The interaction modeling is very rich since all inter-correlations between feature dimensions of the different modalities are considered. However, this quantity of possible interactions limits the dimensions of the projected space, which can cause a bottleneck in the model.

BLOCK being built on the block-term decomposition, we constrain the tensor using a combination of both concepts, which provides a richer modeling of the interactions between modalities. This richness is ensured by the R tensors $\mathcal{D}_r$, each parametrizing a bilinear function that takes as inputs chunks of $\tilde{\boldsymbol{x}}^1$ and $\tilde{\boldsymbol{x}}^2$. As this interaction modelling is done by chunks and not for every possible combination of components in $\tilde{\boldsymbol{x}}^1$ and $\tilde{\boldsymbol{x}}^2$, we can reach high dimensions in the projections $\boldsymbol{A}$ and $\boldsymbol{B}$ without exploding the number of parameters in $\mathcal{D}^{bd}$. This property of having a fine interaction modeling between high dimensional projections is very desirable in our context where we need to model complex interactions between high-level semantic spaces. As we show in the experiments, performance of a bilinear model strongly depends on both the number and the size of the blocks $\mathcal{D}_r$ that parametrize the system.

3 BLOCK fusion for VQA task

The task of Visual Question Answering (Antol et al. 2015), (Goyal et al. 2017) has been a fertile playground for researchers to investigate on how bilinear models should be used. In the classical setup for VQA, shown in Figure 2, image and question have to be merged with a multimodal fusion technique, which can be implemented as an instance of bilinear model. The answer is predicted through a classification layer that follows the fusion module. In MCB (Fukui et al. 2016), the bilinear interaction is simplified using a sketching technique. However, more recent techniques tackle this complexity issue from a tensor decompositions standpoint: MLB (Kim et al. 2017) and MUTAN (Ben-Younes et al. 2017) constrain the tensor of parameters using respectively the CP and Tucker decomposition. In MFB (Yu et al. 2017b), the tensor is viewed as a stack of matrices, and a classical matrix rank constraint is imposed on each of them. Finally in MFH (Yu et al. 2018), multiple MFB blocks are cascaded to model higher-order interactions between inputs. Most recent techniques embed some of these fusion strategies into more elaborated architecture that involve multiple types of visual features (Jiang et al. 2018), or specific modules (Zhang, Hare, and Prgel-Bennett 2018) designed for precise question types. In this section, we compare BLOCK to the other multimodal fusion techniques, and show how it surpasses them both in terms of performance and number of parameters.

3.1 VQA architecture

Our VQA model is based on a classical attentional architecture (Fukui et al. 2016), enriched by our proposed merging scheme. Our fusion model is shown in Figure 2. We use the Bottom-up image features provided by (Teney et al. 2018), consisting of a set of detected objects and their representation (see (Mordan et al. 2017; Durand et al. 2017) for further insights on detection and localization). To get a vector embedding of the question, words are preprocessed and then fed into a pretrained Skip-thought encoder (Kiros et al. 2015). The outputs of this language model are used to produce a single vector representing the whole question, as in (Yu et al. 2018). We use a BLOCK fusion to merge the question and image representations. The question vector is used as a context to guide the visual attention. Saliency scores are produced using a BLOCK fusion between each image vector and the question embedding.

Details: For the BLOCK layers, we set $L = M = N = 80, R = 20$ and constrain the rank of each mode-3 slices of each block to be less than 10. We found these hyperparameters with a a cross-validation on the *val* set. As in (Yu et al. 2018), we consider the 3000 most frequent answers. As in (Ben-Younes et al. 2017), we use a cross-entropy loss with answer sampling. We jointly optimize the parameters of our VQA model using Adam (Kingma and Ba 2015) with a learning rate of $1e^{-4}$, without learning rate decay or gradient clipping, and with a batch size of 200. We early stop the training of our models according to their accuracy on a holdout set.

3.2 Fusion analysis

In Table 1, we compare BLOCK to 8 different fusion schemes available in the literature on the commonly used VQA 2.0 Dataset (Goyal et al. 2017). This dataset is composed of 658,111 image-question-answers triplets for training and validation (*trainval* set), and 447,793 triplets for evaluation (*test-std* set). We train on *trainval* minus a small subset used for early-stopping, and report the performance on *test-dev* set. For each fusion strategy, we run a grid search over its hyperparameters and keep the model that performs best on our validation set. We report the size of the model,

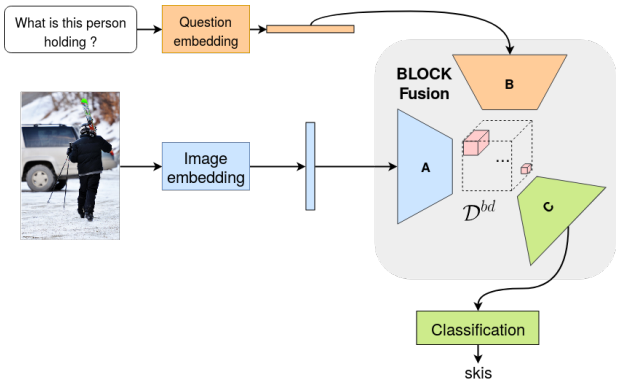


Figure 2: Architecture for VQA that embeds the BLOCK bilinear fusion. To make this system more efficient, we integrate the fusion in an attentional framework.

corresponding to the number of parameters between the attended image features, the question embedding, and the answer prediction. We briefly describe the different fusion schemes used for the comparison:

– (1) the two vectors are projected on a common space, and their summation is projected to predict the answer;

– (2) the vectors are concatenated and passed at the input of a 3-layer MLP;

– (3) a bilinear interaction based on a count-sketching technique that projects the outer product of between inputs on a multimodal space;

– (4) a bilinear interaction where the tensor is expressed as a Tucker decomposition;

– (5) a bilinear interaction where the tensor is expressed as a CP decomposition;

– (6) a bilinear interaction where each 3rd mode slice matrix of the tensor is constrained by its rank;

– (7) a bilinear interaction where the tensor is expressed as a Tucker decomposition, and where its core tensor has the same rank constraint as (6);

– (8) a higher order fusion composed of cascaded (6);

– (9) our BLOCK fusion.

From the results in Table 1, we see that the simple sum fusion (1) provides a very low baseline. We also note that the MLP (2) doesn't provide the best results, despite its non-linear structure. As the MLP should be able to find that two different modalities are used and that it needs to look for interactions between them, this is in practice difficult to obtain. Instead, top performing methods are based on a bilinear model. The structure imposed on the parameters highly influences the final performance. We can see that (3), which simplifies the bilinear model using random projections, has efficiency issues due to the count-sketching technique. These issues are alleviated in the other bilinear methods, which use the tensor decomposition framework to practically implement the interaction. Our BLOCK method (9) gives the best results. As we saw, the block-term decomposition generalizes both CP and Tucker decompositions, which is why it is not surprising to see it surpass them. Moreover, the fact that it integrates the 3rd order slices rank constraint gives it the advantages of (6) and (7). Interestingly, it even surpasses (8) which is based on a higher-order interaction modeling, while using 30M less parameters. This strongly indicates that controlling a bilinear model through its block-term ranks provides an efficient trade-off between modeling capacities and number of parameters. To further validate this hypothesis, we evaluate a BLOCK fusion with only 3M parameters. This model obtained 64.91%. Unsurprisingly, it does not surpasses all the methods against which we compare. However, it obtains competitive results, improving over 5 out of 8 methods that all use far more parameters.

3.3 Comparison to leading VQA methods

We compare our model with state-of-the-art VQA architecture on two datasets: the widely used VQA 2.0 Dataset (Goyal et al. 2017) and TDIUC (Kafle and Kanan 2017). On this more recent dataset, evaluation metrics are provided to assess the robustness of the model with respect to answer imbalance, as well as to account for performance homogeneity across the difference question types.

As we show in Table 2, our model is able to outperform the preceding ones on TDIUC by a large margin for every metrics, especially those which account for bias in the data. We notably report a gain of +1.7 in accuracy, +3.95 in A-MPT, +5.05 in H-MPT, +16.12 in A-NMPT, +15.45 in H-NMPT, over the best scoring model in each metric. The high results in the harmonic metrics (H-MPT and H-NMPT) suggest that BLOCK performs well across all question types, while the high scores in the normalized metrics (A-NMPT and H-NMPT) denote that our model is robust to answer imbalance type of bias in the dataset.

In Table 3, we see that our fusion model obtains competitive results on VQA 2.0 compared to previously published methods. As we are outperformed by (Zhang, Hare, and Prgel-Bennett 2018), whose proposition rely on a completely different architecture, we believe that both our contributions are orthogonal. Still, our model performs better than (Teney et al. 2018) and (Yu et al. 2018), with whom we share the global VQA architecture. In further details, we point out that BLOCK surpasses (Yu et al. 2018) reaching a +1.78 improvement in the overall accuracy on *test-dev*, even though the latter encompasses the current state-of-the-art fusion scheme. Furthermore, we use the same image features than (Teney et al. 2018) and are able to achieve a +2.26 gain on *test-dev* and +2.25 on *test-std*.

4 VRD task

The task of Visual Relationship Detection aims at predicting triplets of the type *"subject-predicate-object"* where *subject* and *object* are localized objects, and *predicate* is a label corresponding to the relationship that links them (for example: "man-riding-bicycle", "woman-holding-phone"). To predict this relationship, multiple types of information are available, for both the subject and object regions: classes, bounding box coordinates, visual features, *etc.* However, this context being more recent than VQA, fusion techniques are less formalized and more *ad-hoc*. In (Hanwang Zhang 2017), the

Table 1: Comparison of the fusion schemes on VQA2 *test-dev* set. $|\Theta|$ is the number of parameters learned in the fusion modeling. *All* is the overall Open Ended accuracy (higher is better). *Yes/no*, *Numbers* and *Others* are subsets that correspond to answers types. In the descriptions, the letter B corresponds to a bilinear model.

	Description	Reference	$\|\Theta\|$	All	Yes/no	Number	Other
(1)	Linear	Sum	8M	58.48	71.89	36.56	52.09
(2)	Non-linear	Concat MLP	13M	63.85	81.34	43.75	53.48
(3)	B + count-sketching	MCB (Fukui et al. 2016)	32M	61.23	79.73	39.13	50.45
(4)	B + Tucker decomp.	Tucker (Ben-Younes et al. 2017)	14M	64.21	81.81	42.28	54.17
(5)	B + CP decomp.	MLB (Kim et al. 2017)	16M	64.88	81.34	43.75	53.48
(6)	B + low-rank on the 3rd mode slices	MFB (Yu et al. 2017a)	24M	65.56	82.35	41.54	56.74
(7)	Combination of (4) and (6)	MUTAN (Ben-Younes et al. 2017)	14M	65.19	82.22	42.1	55.94
(8)	Higher order fusion	MFH (Yu et al. 2018)	48M	65.72	82.82	40.39	56.94
(9)	B + Block-term decomposition	BLOCK	18M	**66.41**	**82.86**	**44.76**	**57.3**

Table 2: State-of-the-art comparison on the TDIUC testing set. * scores reported from (Kafle and Kanan 2017).

Model	Accuracy	A-MPT	H-MPT	A-NMPT	H-NMPT
Most common answer (Kafle and Kanan 2017)	51.15	31.11	17.53	15.63	0.83
Question only (Kafle and Kanan 2017)	62.74	39.31	25.93	21.46	8.42
NMN* (Andreas et al. 2016)	79.56	62.59	51.87	34.00	16.67
MCB* (Fukui et al. 2016)	81.86	67.90	60.47	42.24	27.28
RAU* (Noh and Han 2016)	84.26	67.81	59.00	41.04	23.99
BLOCK	**85.96**	**71.84**	**65.52**	**58.36**	**39.44**

relation is predicted by a substractive fusion between subject and object representations, each consisting in a linear function of relative coordinates, class distributions and visual features. (Li et al. 2017) predicts the relationship by a complex message passing structure between subject and object representations, and (Dai, Zhang, and Lin 2017) uses a formulation inspired from Conditional Random Fields to perform joint recognition between the subject, object and predicate classes. We adopt in the following a very simple architecture, to put emphasis on the fusion module between different information sources.

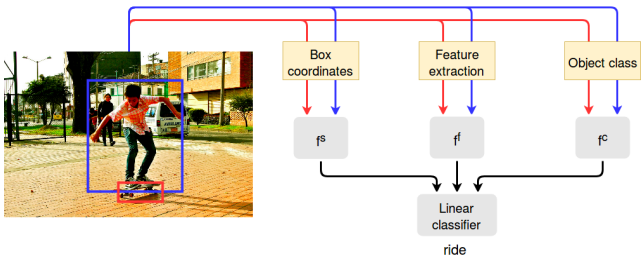


Figure 3: Architecture of our visual relationship detection system

4.1 VRD Architecture

Our VRD architecture is shown in Figure 3. It takes as inputs a subject and an object bounding box. Each of them is represented as their 4-dimensional box spatial coordinates $\boldsymbol{x}_s^s$ and $\boldsymbol{x}_o^s$ (normalized between 0 and 1), their object classes $\boldsymbol{x}_s^c$ and $\boldsymbol{x}_o^c$, and their semantic visual features $\boldsymbol{x}_s^f$ and $\boldsymbol{x}_o^f$. To predict the relationship predicate, we use one fusion module for each type of features following Eq. (10).

$$\boldsymbol{x} = \left[f^s\left(\boldsymbol{x}_s^s, \boldsymbol{x}_o^s\right), f^c\left(\boldsymbol{x}_s^c, \boldsymbol{x}_o^c\right), f^f\left(\boldsymbol{x}_s^f, \boldsymbol{x}_o^f\right)\right] \qquad (10)$$

where f can be implemented as BLOCK, or any other multimodal fusion. Each fusion module outputs a vector of dimension d, all concatenated into a 3d-dimensional vector that will serve as an input to a linear layer predictor $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$. The system is trained with back-propagation on a binary-crossentropy loss.

An other important component is the object detector. As usually done, we first train a Faster-RCNN on the object boxes of the VRD dataset. For Predicate prediction, we use it as a features extractor given the ground truth bounding boxes. For Phrase detection and Relationship detection, we use it to extract the bounding boxes with their associated features.

For a system to perform well on Phrase and Relationship detection, it should have been also trained on pairs of (subject, object) boxes that are not linked together by any relationship. During training, we randomly sample half of all possible *negative pairs*, and assign them an all-zeros label vectors.

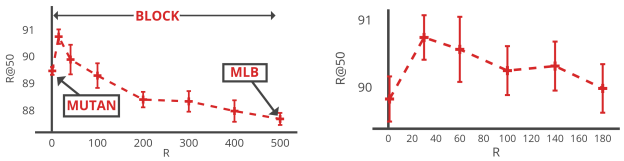
4.2 Dataset

The VRD dataset (Lu et al. 2016) is composed of 5,000 images with 100 object categories and 70 predicates. It contains 37,993 relationships with 6,672 unique triplets and an average of 24.25 predicates per object category. The dataset

Table 3: State-of-the-art results on VQA2 testing sets. The models were trained on the union of VQA 2.0 *trainval* split and VisualGenome (Krishna et al. 2017) train split. *All* is the overall OpenEnded accuracy (higher is better). *Yes/no*, *Numbers* and *Others* are subsets that correspond to answers types. Only single model scores are reported. * scores reported from (Goyal et al. 2017)

Model	VQA2 Test-dev				VQA2 Test-std			
	All	Yes/no	Num.	Other	All	Yes/no	Num.	Other
Most common answer (Goyal et al. 2017)	-	-	-	-	25.98	61.20	0.36	1.17
Question only (Goyal et al. 2017)	-	-	-	-	44.26	67.01	31.55	27.37
Deep LSTM* (Lu et al. 2015)	-	-	-	-	54.22	73.46	35.18	41.83
MCB* (Fukui et al. 2016)	-	-	-	-	62.27	78.82	38.28	53.36
ReasonNet (Ilievski and Feng 2017)	-	-	-	-	64.61	78.86	41.98	57.39
TipsAndTricks (Teney et al. 2018)	65.32	81.82	44.21	56.05	65.67	82.20	43.90	56.26
MFH (Yu et al. 2018)	65.80	-	-	-	-	-	-	-
Counter (Zhang, Hare, and Prgel-Bennett 2018)	**68.09**	83.14	**51.62**	**58.97**	**68.41**	83.56	**51.39**	**59.11**
BLOCK	67.58	**83.6**	47.33	58.51	67.92	**83.98**	46.77	58.79

is divided between 4,000 images for training and 1,000 for testing. Three different settings are commonly used to evaluate a model on VRD: (1) **Predicate prediction**: the coordinates and class labels are given for both subject and object regions. This setup allows to assess the model's ability to predict a relationship, regardless of the object detection stage. (2) **Phrase detection**: a predicted triplet $<$*subject, predicate, object*$>$ matches a ground-truth if the three labels match and if the union region of its bounding boxes matches the union region of the ground-truth triplet, with IoU above 0.5. (3) **Relationship detection**: more challenging than (2), this one requires that both subject and object intersect with an IoU higher than 0.5. For each of these settings, performance is usually measured with Recall@50 and Recall@100



(a) (b)

Figure 4: Performance of BLOCK with respect to the structure of the block-superdiagonal tensor. Scores are given on a holdout validation set. 4a: The size of the core tensor is fixed to $RL = RM = RN = 500$. 4b: The total number of parameters in the block-diagonal tensor is fixed to $555K$.

4.3 Fusion analysis

To show the effectiveness of the BLOCK bilinear fusion, we run the same type of experiment we did in the previous section. For each fusion technique, we use the architecture described in Equation 10 where we replace f by the corresponding bilinear function. As we did for VQA, we cross-validate the hyperparameters of each fusion technique and keep the best model each time. In Table 4, we see that BLOCK still outperforms all previous methods on each of the three tasks. We can remark that for this task, the non linear MLP perform relatively well compared to the other methods. It is likely that an MLP can model the interactions at stake for VRD more easily than those for VQA. However, we can improve over this strong baseline using a BLOCK fusion.

In the next experiments, we validate the power of our BLOCK fusion, and analyze how it behaves under different setups. We randomly split the training set into three train/val sets, and plot the mean and standard deviation of the recall@50 calculated over them. In Figure 4a, we fix the dimension of the block-superdiagonal tensor to $RL = RM = RN = 500$ and vary the number of blocks used to fill this tensor. When $R = 1$, which corresponds to the Tucker decomposition, the number of parameters in the core tensor is equal to $500^3 = 125M$, making the system arduously trainable on our dataset. On the opposite, when $R = 500$, the number of parameters is controlled, but the mono-modal projections are only allowed to interact through an element-wise multiplication, which makes the interaction modeling relatively poor. The block-term decomposition provides an in-between working regime, reaching an optimum when $R \approx 20$.

In Figure 4b, we keep the number of parameters fixed. As the number of chunks increases, the dimensions of the mono-modal projections also increases. Once again, an optimum is reached when $R \approx 20$. These results confirm our hypothesis that the way the parameters are distributed within the tensor, in terms of size and number of blocks, has a real impact on the system's performance.

4.4 Comparison to leading VRD methods

In Table 5, we compare our system to the state-of-the-art methods on VRD. On predicate prediction, our fusion outperforms all previous methods on R@50, including (Yu et al. 2017a) that uses external data. On R@100, the BLOCK fusion is only marginally outperformed by (Yu et al. 2017a),

Table 4: Comparative study of the different multimodal fusion strategies on the VRD test-set. The reported metrics are the Recall@K in %.

	Description	$\|\Theta\|$	Predicate Prediction		Phrase Detection		Relationship Detection	
			R@50	R@100	R@50	R@100	R@50	R@100
(1)	Linear	2.5M	82.99	89.68	14.44	16.94	9.73	11.34
(2)	Non-linear	2M	84.47	91.6	21.9	24.69	15.79	17.83
(3)	B + count-sketching	2M	82.23	89.07	13.42	15.8	9.17	10.79
(4)	B + Tucker decomp.	3M	83.25	89.77	11.23	14.09	7.37	9.00
(5)	B + CP decomp.	4M	85.96	91.66	23.67	26.50	16.41	18.59
(6)	B + low-rank on the 3rd mode slices	15M	85.21	91.06	25.31	28.03	17.83	19.77
(7)	Combination of (4) and (6)	30M	85.65	91.33	25.77	28.65	18.53	20.38
(8)	Higher order fusion	16M	85.58	91.3	26.09	28.73	18.81	20.63
(9)	Block-term decomposition	5M	**86.58**	**92.58**	**26.32**	**28.96**	**19.06**	**20.96**

Table 5: State-of-the-art results on the VRD testing set. The reported metrics are the Recall@K in %.

Model	External data	Predicate Prediction		Phrase Detection		Relationship Detection	
		R@50	R@100	R@50	R@100	R@50	R@100
Yu et. al (Yu et al. 2017a)	✓	85.64	94.65	26.32	29.43	22.68	31.89
Li et. al (Li et al. 2017)	✗	-	-	22.78	27.91	17.32	20.01
Liang et. al (Liang, Lee, and Xing 2017)	✗	-	-	21.37	22.60	18.19	20.79
Zhang et. al (Hanwang Zhang 2017)	✗	44.76	44.76	19.42	22.42	14.07	15.20
Lu et. al (Lu et al. 2016)	✗	47.87	47.87	16.17	17.03	13.86	14.70
Peyre et. al (Peyre et al. 2017)	✗	52.6	52.6	17.9	19.5	15.8	17.1
Dai et. al (Dai, Zhang, and Lin 2017)	✗	80.78	81.90	19.93	23.45	17.73	20.88
BLOCK	✗	**86.58**	**92.58**	**26.32**	**28.96**	**19.06**	**20.96**

but we perform better than all methods that don't use extra data. These results validate the efficiency of the block-term decomposition to predict a predicate by fusing information coming from ground truth subject and object boxes. On phrase detection, our BLOCK fusion achieves better results than all previous models in R@50. Notably, the scores obtained for phrase detection are lower than for predicate prediction, since the ground truth regions are not provided in this setup. Finally, on relationship detection, BLOCK surpasses all previous methods without extra data in R@50, and gives similar performance than (Dai, Zhang, and Lin 2017) in R@100. The scores for relationship detection are lower than for phrase detection: in this setup, a prediction is positive if both subject and object boxes match the ground truth. On contrary, in phrase detection, the comparison between prediction and ground truth is done on the union between subject and object regions. Lastly, unlike some of the methods reported in Table 5, we do not fine-tune or adapt the detection network to the visual relationship tasks.

5 Conclusion

In this work, we introduce BLOCK, a bilinear fusion model whose tensor of parameters is structured using the block-term decomposition. BLOCK aims at optimizing the trade-off between complexity and modeling capacities, and combines the strengths of the CP and Tucker decompositions. It offers the possibility to model rich interactions between groups of features, while still using high-dimensional monomodal representations.

We apply BLOCK for two challenging computer vision tasks: VQA and VRD, where the parameters of our BLOCK fusion model are learned. Comparative experiments show that BLOCK improves over previous fusion schemes including linear, bilinear and non-linear models. We also show that BLOCK is able to maintain competitive performances with very compact parametrization.

In future works, we plan to extend the BLOCK idea to other applications. In particular, we want to explore the use of multiple input and output modalities, and to apply BLOCK for interpreting and explaining the behaviour of the multimodal deep fusion model (Engilberge et al. 2018; Carvalho et al. 2018).

Acknowledgements This work has been supported within the Labex SMART supported by French state funds managed by the ANR within the Investissements dAvenir programme under reference ANR-11-LABX-65.

References

Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D. 2016. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 39–48.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. VQA: Visual Question Answering. In *ICCV*.

Ben-Younes, H.; Cadène, R.; Thome, N.; and Cord, M. 2017. Mutan: Multimodal tucker fusion for visual question answering. *ICCV*.

Carroll, J. D., and Chang, J.-J. 1970. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*.

Carvalho, M.; Cadène, R.; Picard, D.; Soulier, L.; Thome, N.; and Cord, M. 2018. Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings. In *SIGIR*.

Cichocki, A.; Mandic, D. P.; Phan, A. H.; Caiafa, C. F.; Zhou, G.; Zhao, Q.; and Lathauwer, L. D. 2015. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*.

Dai, B.; Zhang, Y.; and Lin, D. 2017. Detecting visual relationships with deep relational networks. In *CVPR*.

De Lathauwer, L. 2008. Decompositions of a higher-order tensor in block terms — part ii: Definitions and uniqueness. *SIAM J. Matrix Anal. Appl.* 30(3):1033–1066.

Duong, C. T.; Lebret, R.; and Aberer, K. 2017. Multimodal classification for analysing social media. *ECML-PKDD*.

Durand, T.; Mordan, T.; Thome, N.; and Cord, M. 2017. WILDCAT: weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *CVPR*.

Engilberge, M.; Chevallier, L.; Pérez, P.; and Cord, M. 2018. Finding beans in burgers: Deep semantic-visual embedding with localization. In *CVPR*.

Fukui, A.; Park, D. H.; Yang, D.; Rohrbach, A.; Darrell, T.; and Rohrbach, M. 2016. In *EMNLP*.

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*.

Hanwang Zhang, Zawlin Kyaw, S.-F. C. T.-S. C. 2017. Visual translation embedding network for visual relation detection. In *CVPR*.

Harshman, R. A.; Ladefoged, P.; Reichenbach, H.; Jennrich, R. I.; Terbeek, D.; Cooper, L.; Comrey, A.; Bentler, P. M.; Yamane, J.; Vaughan, D.; and Jahnke, B. 2001. Foundations of the parafac procedure: Models and conditions for an "explanatory" multimodal factor analysis.

Ilievski, I., and Feng, J. 2017. Multimodal learning and reasoning for visual question answering. In *Advances in Neural Information Processing Systems*, 551–562.

Jiang, Y.; Natarajan, V.; Chen, X.; Rohrbach, M.; Batra, D.; and Parikh, D. 2018. Pythia v0.1: The winning entry to the vqa challenge 2018. `https://github.com/facebookresearch/pythia`.

Kafle, K., and Kanan, C. 2017. An analysis of visual question answering algorithms. In *The IEEE International Conference on Computer Vision (ICCV)*.

Kim, J.-H.; On, K. W.; Lim, W.; Kim, J.; Ha, J.-W.; and Zhang, B.-T. 2017. Hadamard Product for Low-rank Bilinear Pooling. In *The 5th International Conference on Learning Representations*.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kiros, R.; Zhu, Y.; Salakhutdinov, R.; Zemel, R. S.; Torralba, A.; Urtasun, R.; and Fidler, S. 2015. Skip-thought vectors. In *NIPS*.

Kiros, R.; Salakhutdinov, R.; and Zemel, R. S. 2015. Unifying visual-semantic embeddings with multimodal neural language models. *TACL*.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision* 123(1):32–73.

Li, Y.; Ouyang, W.; Wang, X.; and Tang, X. 2017. Vip-cnn: Visual phrase guided convolutional neural network. In *CVPR*.

Liang, X.; Lee, L.; and Xing, E. P. 2017. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *CVPR*.

Lu, J.; Lin, X.; Batra, D.; and Parikh, D. 2015. Deeper lstm and normalized cnn visual question answering model.

Lu, C.; Krishna, R.; Bernstein, M.; and Fei-Fei, L. 2016. Visual relationship detection with language priors. In *ECCV*.

Mordan, T.; Thome, N.; Henaff, G.; and Cord, M. 2017. Deformable part-based fully convolutional network for object detection. In *BMVC*.

Noh, H., and Han, B. 2016. Training recurrent answering units with joint loss minimization for vqa. *arXiv preprint arXiv:1606.03647*.

Peyre, J.; Laptev, I.; Schmid, C.; and Sivic, J. 2017. Weakly-supervised learning of visual relations. *ICCV*.

Teney, D.; Anderson, P.; He, X.; and van den Hengel, A. 2018. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tucker, L. R. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika* 31(3):279–311.

Yu, R.; Li, A.; Morariu, V. I.; and Davis, L. S. 2017a. Visual relationship detection with internal and external linguistic knowledge distillation. In *ICCV*.

Yu, Z.; Yu, J.; Fan, J.; and Tao, D. 2017b. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. *ICCV*.

Yu, Z.; Yu, J.; Xiang, C.; Fan, J.; and Tao, D. 2018. Beyond bilinear: Generalized multi-modal factorized high-order pooling for visual question answering. *IEEE TNNLS*.

Zhang, Y.; Hare, J.; and Prgel-Bennett, A. 2018. Learning to count objects in natural images for visual question answering. In *ICLR*.