# Binary Visual Question Answering using Transformers with raw inputs

Abdulrahman Alabdulkareem

arkareem@mit.edu

## Abstract

*In this paper we introduce the Visual Question Answering task and the balanced binary visual question answering dataset our work utilizes. We propose two models, one that is used as a baseline model which is a latent Joint-Embedding model that utilizes Transformer networks to embed the visual and textual parts of the question. We then propose our main model which is an attention model that also utilizes transformer networks as backbone and is able to achieve relatively good results and beats our baseline latent Joint-Embedding model with the added benefit of being able to see the attention mask to visualize where the model is looking with respect to the question. Finally, we provide, in the supplemental material, visualizations of our model applied to the test set which shows which parts of the image the model is looking at to answer the question.*

## 1. Introduction

Visual Question Answering (VQA) is a task that combines Computer Vision and Natural Language Processing that tests the capability of answering open-ended questions based on a provided image. This is an interesting task as it requires the machine to learn complicated concepts from both vision and language and combine them in a coherent manner to give a proper answer. Many datasets have been created as benchmarks for this task [1, 6, 12, 13].

We will be training and testing our proposed model on the balanced binary visual question answering dataset proposed in [20] which is a modified subset of the dataset proposed in [1] describe in Sec. 2.1.

## 2. Related Work

A survey on VQA models and datasets is done in [18] where they describe the different dataests and approaches in the literature. To solve this problem, the machine needs to somehow combine the textual information in the question to the visual information in the image which is what separates the different approaches to this problem, these approaches include Joint Embedding, attention mechanisms, compositional models, etc. [18].

The approach used by [20] to solve their own dataset is a classical attention mechanisms approach Where they utilize a classical NLP approach using the Stanford parser to parse the question for the primary (P) and secondary (S) objects along with their relation (R) to end up with a $< P, R, S >$ tuple. Then, they utilize the provided scene features along with a method proposed in [8] to focus on parts of the scene that are represented by the tuple $< P, R, S >$ then encode the visual representation of the those attended sections.

Further work has been done by [14] on the same dataset achieving better results by using Graph Neural Networks followed by an attention mechanism. As with [20] they utilized the Stanford parser to parse the question into a graph where the nodes are word ids and the edges are types of dependencies. They then use the scene features and represent them as a graph of objects and properties followed by passing both graphs to a Graph Neural Network followed by an attention mechanism to produce an output.

### 2.1. Dataset

The dataset introduced in [1] is a dataset for Visual Question Answering that contains images paired with an open-ended question and their answer. The images are composed of real images and abstract scenes, the real images subset of the dataset needs a complex and noisy object detectors to analyze the scene before the reasoning part of the question answering can take place. Thus abstract scenes were introduced in the dataset that are scenes generated in a toy world such that each image in this toy world is accompanied by a feature vector perfectly describing the scene.

### 2.2. Imbalanced Dataset conditioned on the question

As pointed out in [20], a problem in the VQA dataset is that strong language priors can achieve surprisingly good results while completely ignoring the image part of the input. For example, They showed how ”tennis” is the answer to 41% of questions asking about the type of sport in the image.

Focusing on the subset of VQA with ”yes” or ”no” answers, this problem is still present. As stated in [20], a Neural Network can achieve an accuracy of 78% despite ignoring the image. This problem arises from the fact that the probability of an answer conditioned on only the question is imbalanced (not uniform). Concretely:

$$P(answer = "yes"|question) \neq 0.5 \tag{1}$$

and

$$P(answer = "no"|question) \neq 0.5 \tag{2}$$

This imbalance means that models can achieve a higher than 50% accuracy by ignoring the image and finding clever priors in the question which then fail to generalize to the validation set. Many past works have demonstrated the importance of a balanced dataset in learning [4, 7, 10, 16]

### 2.3. Balancing VQA

Perfectly balancing the "yes"/"no" subset of the dataset conditioned on the question may seem like an impossible task as this would mean that

$$\begin{aligned} &P(answer = "yes"|f(question)) \\ =&P(answer = "no"|f(question)) \\ =&0.5 \end{aligned} \tag{3}$$

For any arbitrary function f. However, this is where one of the major advantages of generated abstract scenes can be utilized.

For each question-image pair, Zhang *et al*. [20] created a complementary scene that is minimally edited such that for the same question the edited image gives the opposite answer. This was only possible by using the fact that abstract scenes can be edited by humans, the end result is that each question has two very similar images which produce opposite answers. This produces a perfectly balanced dataset which fulfils Eq. (3).

### 2.4. Scene Features

Since the images in the dataset are generated using a composition of clip arts, the authors have added an extra set of features for each input which is the underlying data used to generate the scene. Scene features include the ID's of all the objects in the scene along with their coordinates in the image space and any modifications or deformations on that objects where all of these are described in numbers. These scene features can perfectly describe the image in a significantly lower dimensional space compared to the raw RGB input.

A significant departure from the methodology used in [14, 20] is that we have opted not to use the scene features and instead rely solely on the raw RGB image for the visual representation of the scene. Operating on the image which is in a much higher dimension is a much more challenging task. We have chosen to do this as this is more closely to how humans would solve this task, in addition to the fact that such features would not be present when dealing with real world images.

### 2.5. Data Leakage

The Abstract Scenes Dataset [2] can be used for extra training examples as it offers 20k/10k training and validation examples which we have utilized in our training. However, special care needs to be taken as the balanced binary validation set provided in [20] shares some of the images in the validation set of [2]. Thus, special care needs to be taken not to accidentally train on them which will cause a data leakage and result in a model that achieves a deceivingly high accuracy.

## 3. Approach

We propose two different approaches to this problem, the first is to use a Joint Embedding approach first used in image-captioning as mentioned in [18]. While the second approach is an attention mechanism approach.

The first approach, at a high level, is to extract the visual and textual features from the image and question respectively. Then, mapping them to a common space where we combine them and map the combined embedding to an output.

### 3.1. Joint Embedding

To mathematically describe our Joint Embedding approach let us first define some notation on the dataset. Let a data point and label $(x^{(i)}, y^{(i)}) \in D$ be defined as $x^{(i)} = (x_v^{(i)}, x_q^{(i)})$ where $x_v^{(i)}$ is an RGB image of an abstract scene and $x_q^{(i)}$ is a question corresponding to that scene, and $y^{(i)} \in \{"yes", "no"\}$ is the binary answer to $x_q^{(i)}$ when asked on the scene $x_v^{(i)}$.

Our approach is to extract the visual features and textual features as

$$v^{(i)} = F_v(x_v^{(i)}) \quad \text{and} \quad q^{(i)} = F_q(x_q^{(i)}) \tag{4}$$

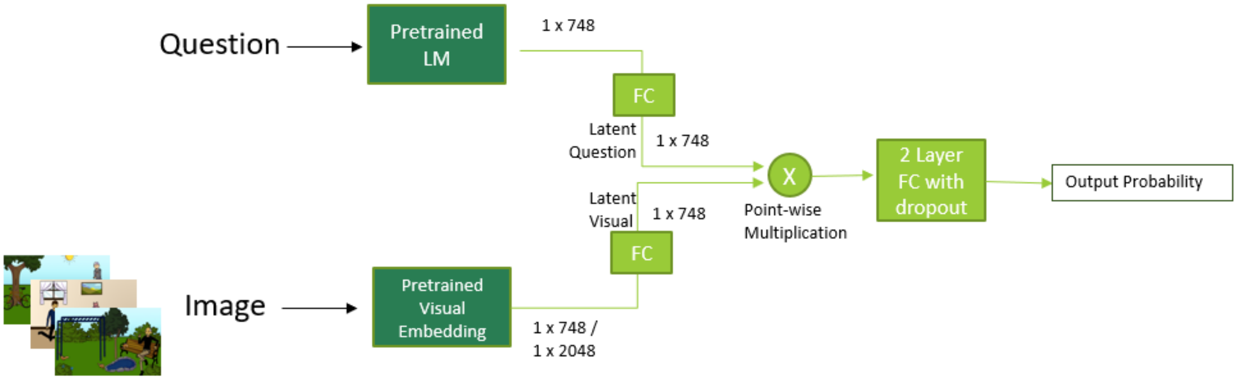
Where $F_v$ extracts the visual features of the image as an $n_v$ dimensional vector and $F_q$ extracts the textual features of the question as an $n_q$ dimensional vector

$$v^{(i)} \in \mathbb{R}^{n_v} \quad \text{and} \quad q^{(i)} \in \mathbb{R}^{n_q} \tag{5}$$

Then we linearly project the two feature vectors to a common $n_c$ dimensional space and pass the result through an activation layer

$$x_v^{(i)} = \sigma(W_v * v^{(i)}) \quad \text{and} \quad x_q^{(i)} = \sigma(W_q * q^{(i)}) \tag{6}$$

where the $W_v \in \mathbb{R}^{n_c \times n_v}$ and $W_q \in \mathbb{R}^{n_c \times n_q}$, and the we use ReLU for the activation layer $\sigma$. Then we take



(a)

Figure 1. A visualization of our Latent Joint-Embedding model.

the Hadamard product of the two projected feature vectors (element-wise multiplication) and project that to a single number passed to a sigmoid to get a probability

$$P(y^{(i)} = ''yes'') = \bar{y} = \sigma(W_o * (x_v^{(i)} \odot x_q^{(i)})) \qquad (7)$$

Where $W_o \in \mathbb{R}^{1 \times n_c}$.

### 3.2. Visual Embedding

For the extraction of visual embedding $F_v(\cdot)$, we have tried several different pre-trained models. Below we listed the two that achieved the best results

**Resnext101 32x8d** is a CNN model developed and pre-trained by Facebook in 2018 [9]. The **32x32d** model achieves 97.6% on the top-5 ImageNet-1k validation set which sets it as #20 in the leaderboard and was considered state of the art in 2018. The **32x32d** model was too large for us to use so we opted for the much smaller **32x8d** model which is about a fourth of the size. We use the CNN as a feature extractor by removing the last classification layer.

**Vit Base Patch16 224 in21k** is a Vision Transformer (ViT) model developed by Google Research introduced in 2020 [17]. Out of all the different methods for visual embedding extraction, ViT achieves the best results.

### 3.3. Textual Embedding

For the extraction of textual embedding $F_q(\cdot)$, we found that the best models are Transformer models.

**BERT** is a Language Representation Transformer model developed by Google [5] in 2018 which obtained 11 state of the arts results on natural language processing (NLP) tasks when it released and widely revolutionized the field of NLP [19].

**Sentance-BERT (SBERT)** is a modification of the pre-trained BERT model such that it outputs sentance embedding whereas sentances with similar meaning output embeddings that are close to each other when measured with cosine-similarity [11]. We were able to achieve slightly better results when using SBERT which would make sense as this model was specifically tuned to extract embeddings from sentences.

### 3.4. Reproducibility of Joint Embedding

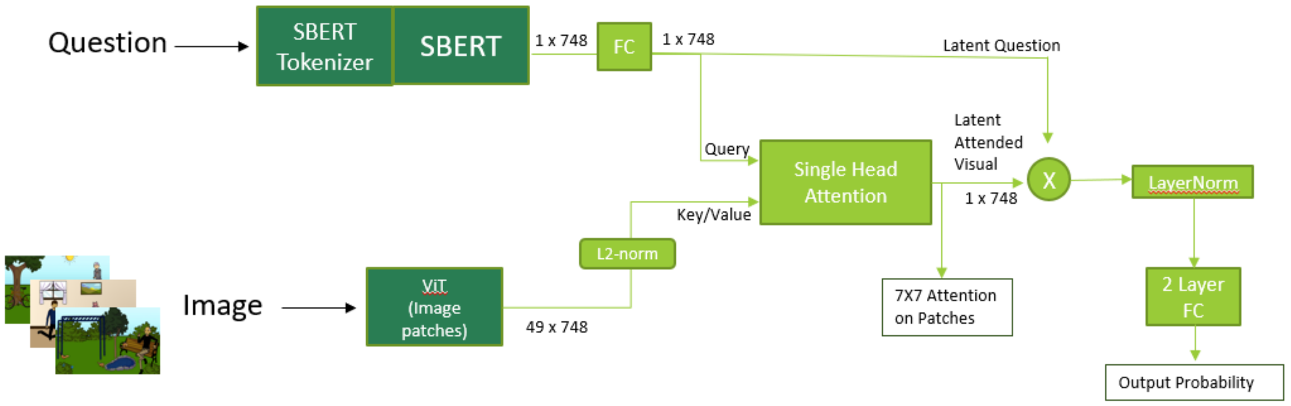
For both models listed in Sec. 3.2 and Sec. 3.3, we have used the basic pre-trained model provided by the Huggingface library and we take the embedding provided by the $(pooler\_output)$ parameter. Then, the linear projection described in Equation (6) is 748 dimensional projection followed by a dropout layer with $p = 0.5$ followed by a 256 dimensional linear projection followed by another dropout layer with $p = 0.5$ then finally projecting that to a single number with a sigmoid activation. We use Binary Cross Entropy loss and an Adam optimizer with learning rate of 1e-4 and train with early stopping for 200 epochs.

## 4. Attention Mechanism Approach

For our attention mechanism approach, we use a similar idea to the Joint Embedding with an addition to an attention mechanism similarly to what is done in [15]. The basic idea of our attention mechanism is to embed the input image into several patches (we use 7x7 patches) instead of just one embedding vector as in Section 3.2. Then we extract the textual embedding by using the same approach as Section 3.3. Then we use the textual embedding to attend to specific visual patches then we pass the output to a linear layer.

### 4.1. Visual Embedding

For the visual embedding, we use the same ViT model mentioned in Section 3.2. However, instead of obtaining a single embed for the entire image, we obtain 49 vector



(a)

Figure 2. A visualization of our final transformer model.

embeddings that represent the embeddings of each of the $7 \times 7$ patches of the image.

$$F_v(x_v^{(i)}) = (v_1^{(i)}, v_2^{(i)}, \ldots, v_{49}^{(i)}) \tag{8}$$

Where $v^{(i)} \in \mathbb{R}^{n_v}$.

### 4.2. Attention Mechanism

After obtaining the embeddings for the visual and texual part of the input, we can perform the attention mechanism which is similar to [15]. The "Query" part of the attention is the textual embedding $q^{(i)}$ (which is a sequence of length 1) and the "Key" and "Value" part of the attention is the visual embedding $(v_1^{(i)}, v_2^{(i)}, \ldots, v_{49}^{(i)})$ (which is a sequence of length 49). The attention described mathematically is as follows:

$$attention\_mask^{(i)} = softmax\left(\frac{(Q^{(i)}W^Q) * (K^{(i)}W^K)^T}{\sqrt{n_{att}}}\right) \tag{9}$$

Where $Q^{(i)} = F_q(x_q^{(i)}) \in \mathbb{R}^{n_q}$ is the query matrix and $K^{(i)} = F_v(x_v^{(i)}) \in \mathbb{R}^{49 \times n_v}$ is the key matrix. And the Query and Key projections are parameterized by $W^Q \in \mathbb{R}^{n_q \times n_{att}}$ and $W^K \in \mathbb{R}^{n_v \times n_{att}}$. Where $n_{att}$ is a parameter that determines the inner-model dimension. Thus, the dimension of $attention\_mask^{(i)} \in \mathbb{R}^{49}$ is a vector of length 49 that sums to 1 and it represents how much to attend to each patch of the input image. Then we use the mask to attend to the visual patches as

$$attention\_out^{(i)} = attention\_mask^{(i)} * (V^{(i)}W^V) \tag{10}$$

Where $V^{(i)} = F_v(x_v^{(i)}) \in \mathbb{R}^{49 \times n_v}$ is the value matrix (which is the same as the key matrix) and it's projection is paramtrized by $W^V \in \mathbb{R}^{n_v \times n_{att}}$. Thus, the output is dimension $attention\_out^{(i)} \in \mathbb{R}^{n_{att}}$. We notice that $attention\_out^{(i)}$ as defined above only incorporates the visual features of the image (attending to specific parts based on the question) but it doesn't exactly incorporate the actual question. To alleviate this, we add the question query to the output of the attention layer then pass that to a fully connected network with 1 hidden layer

$$\begin{aligned} x_{attended}^{(i)} &= attention\_out^{(i)} + (Q^{(i)}W^Q) \\ P(y^{(i)} &= "yes") = \\ &Sigmoid(W^{o2} * ReLU(W^{o1} * x_{attended}^{(i)})) \end{aligned} \tag{11}$$

Where $W^{o1} \in \mathbb{R}^{n_{inner} \times n_{att}}$ and $W^{o2} \in \mathbb{R}^{1 \times n_{inner}}$.

### 4.3. Normalization

We add two normalization layers in our network. The first is an $L2-norm$ applied on each channel individually on the output of the ViT model such that each channel of the 748 channels has an L2 norm of 1. We do this because we noticed that some channels have large activation values while others have very small activation values and when we added the $L2-norm$ layer, the network performed better.

The second normalization layer is a $LayerNorm$ [3] which is usually used on the output of transformers. It normalizes over all hidden units is a single layer to overcome the covariate shift problem by fixing the mean and variance of the summed input [3].

### 4.4. Reproducibility of Attention Model

For both models the visual and textual models we have used the basic pre-trained model provided by the Huggingface library similar to Section 3.4. Then, the attention head

| Model | Test Acc. |
|---|---|
| **Our Latent Model** | 70.34 |
| Q+Tuple+A-IMG‡ [20] | 71.03 |
| **Our Attention Model** | **73.97** |
| Q+Tuple+H-IMG‡ [20] | 74.65 |
| Graph VQA‡ [14] | 74.94 |

Table 1. Results on the test set. Accuracies are in percents using VQA Accuracy Metric [1]
‡ Model utilizes scene features instead of raw RGB image

dimension ($n_{att}$) in Equation (9) is set to 512 and we add a layerNorm after Equation (10). The inner dimension for the fully connected layer at Equation (11) is set to 256 where the hidden layer is followed by a dropout layer with p = 0.5. We use Binary Cross Entropy loss and an Adam optimizer with learning rate of 1e-4 and train with early stopping for 200 epochs.

## 5. Evaluation

We display the final accuracy on the test set for both our best latent Joint-Embedding model and best attention model in Table 1 (best model chosen based on the validation split from the training data). Our attention model achieves very close results to other past works despite the fact that the other models from previous works utilize extra features as input instead of solely relying on the raw RGB image like we do (refer to Section 2.4).

We have further analysis on how the model performs on testing images in the supplementary material where we show the attention mask (from Equation (9)) as a heat map overlaid on top of the image to visualize where the model is looking to answer the question which makes it possible to know if the model answered the question based on luck or if it properly looked at the object of interest.

We can see that the model is able to answer simple questions that are easy to visualize such as ”Is it sunny?” or ”Is there a paining on the wall?” and it looks directly at the locations of interest. All images displayed are from the testing set where the model has never actually looked at. However, we see that the model is unable to answer questions that requires a bit more knowledge of objects such as ”Is the basket open?” due to the lack of training images containing questions about ”baskets”. It also fails to answer questions that require complicated knowledge from multiple parts of the image and their relations such as ”Are the men facing each other?”.

## 6. Conclusion

In conclusion, we present an attention model that answers question by looking at specific parts of the image depending on the question. Our model is comparable to other models from past-works despite the fact that it looks at only the image with no access to the underlying latent variables of the scene.

It is important to be able to identify why the model took the decisions it chose which is especially important for a yes/no task, where the output is a single probability which makes the decision process hard to interpret. Our model has the added benefit of outputting an attention mask that can be used to visualize which parts of the image the model is looking at which gives much better insight on how the model is making decisions. Which is a great tool to see what the model is capable of and its limitations of understanding the scene and complex object relations.

For future work, we propose that our attention model is trained on more data as transformer models are notoriously *”data-hungry”*. Another point of improvement is adding more attention heads instead of using a single attention head to be able to have different heads learn to attend to different parts of the question to able to tackle more complex question.

## References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 1, 5

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[4] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004. 2

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

[6] Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. Are you talking to a machine? dataset and methods for multilingual image question. *Advances in neural information processing systems*, 28, 2015. 1

[7] Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas, et al. Handling imbalanced datasets: A review. *GESTS international transactions on computer science and engineering*, 30(1):25–36, 2006. 2

[8] Xiao Lin and Devi Parikh. Don't just listen, use your imagination: Leveraging visual common sense for non-visual tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1

[9] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018. 3

[10] M Mostafizur Rahman and Darryl N Davis. Addressing the class imbalance problem in medical datasets. *International Journal of Machine Learning and Computing*, 3(2):224, 2013. 2

[11] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019. 3

[12] Andrew Shin, Yoshitaka Ushiku, and Tatsuya Harada. The color of the cat is gray: 1 million full-sentences visual question answering (fsvqa). *arXiv preprint arXiv:1609.06657*, 2016. 1

[13] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. MovieQA: Understanding Stories in Movies through Question-Answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[14] Damien Teney, Lingqiao Liu, and Anton van Den Hengel. Graph-structured representations for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2017. 1, 2, 5

[15] 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. 3, 4

[16] Qiong Wei and Roland L Dunbrack Jr. The role of balanced training and testing data sets for binary classifiers in bioinformatics. *PloS one*, 8(7):e67863, 2013. 2

[17] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision, 2020. 3

[18] Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40, 2017. 1, 2

[19] Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. Pretrained transformers for text ranking: Bert and beyond. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 1154–1156, New York, NY, USA, 2021. Association for Computing Machinery. 3

[20] Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and yang: Balancing and answering binary visual questions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5014–5022, 2016. 1, 2, 5