Supplementary Material: Spatially Aware Multimodal Transformers for TextVQA

A Training and Model Parameters:

All the 6-layer models have 96.6 million parameters and the 4-layer models have 82.4 million parameters. We train our models using Adam optimizer [3] with a linear warmup and with a learning rate of 1e-4 and a staircase learning rate schedule, where we multiply the learning rate by 0.1 at 14000 and at 19000 iterations. We train for 36.1K total iterations (100 epochs) on 2 NVIDIA Titan XP GPUs for 12 hours and use a batch-size of 96 and $d = 768$ as dimensionality for encoding all multi-modal features. We use the PyTorch [4] deep-learning framework for all the experiments.

We list the hyper-parameters used in our experiments for both SA-M4C and M4C models in Table A. We keep these hyper-parameters fixed across all the ablations for both TextVQA [5] and STVQA [1] datasets.

Table A: Hyperparameter choices for models.

#	Hyperparameters	Value	#	Hyperparameters	Value
1	Maximum question tokens	20	2	Maximum object tokens	100
3	Maximum OCR tokens	50	4	Maximum decoding steps	12
5	Embedding size	768	6	Number of Multimodal layers	6N/2N→4S
7	Multimodal layer intermediate size	3072	8	Number of attention heads	12
9	Types of spatial relationships	12	10	Multimodal layer dropout	0.1
11	Context size	1/2	12	Optimizer	Adam
13	Batch size	128	14	Base Learning rate	1e-4
15	Warm-up learning rate factor	0.2	16	Warm-up iterations	1000
17	Vocabulary size	5000	18	Gradient clipping (L-2 Norm)	0.25
19	Number of epochs	100	20	Learning rate decay	0.1
21	Learning rate decay steps	14000, 19000	22	Number of iterations	36000

B List of spatial-prepositions

We used the following list of spatial prepositions to form the subset of questions that involve spatial reasoning: north, south, east, west, up, down, left, right, under, top, bottom, middle, center, above, below, beside, beneath.

C Ablations with varying Spatial Layers

We also study the affect of using spatially aware self-attention layers in a multi-modal transformer. We gradually start replacing the self-attention layers of M4C with our spatially aware self-attention layers. We observe from Table B that, as we replace more layers, the performance gradually increases. However, it is important to keep a couple of normal self-attention layers at the bottom to allow different modalities to attend to the entire context available to them. Since the

spatially aware self-attention layers do not modify the question representations, the self-attention layers in the bottom allow the question tokens to attend to other question tokens as well as object and OCR tokens. Indeed, we see a significant drop as we remove self-attention layers from the bottom.

Table B: Ablations with varying number of spatially aware self-attention layers.

	Method	Model Structure	Context	Accuracy on Val.
1	M4C [2]†	6N	-	42.70
2	SA-M4C	5N→1S	1	42.61
3	SA-M4C	4N→2S	1	43.19
4	SA-M4C	3N→3S	1	43.16
5	SA-M4C	2N→4S	1	**43.80**
6	SA-M4C	1N→5S	1	43.07

D Deforming/Reversing Spatial Graph during Inference

To understand the role of the spatial graph in our approach, using our best model (SA-M4C), we experiment by modifying the spatial graph during inference. For this, we reverse every edge type in the spatial graph (Table C, `Row3`: SA-M4C Rev). For instance, the relationship $\langle \text{obj}_1 - right - \text{obj}_2 \rangle$ now becomes $\langle \text{obj}_1 - left - \text{obj}_2 \rangle$. Similarly, we also experiment by randomly perturbing the spatial graph (Table C, `Row-4`: SA-M4C Rand). For this, we replace each existing relationship between two objects with a random one. We observe a significant performance drop in both the experiments which emphasizes the importance of encoding the spatial relations correctly.

Table C: Effect of randomizing and reversing spatial graph during inference.

	Method	Model Structure	Context	Spatial Graph	w/ ST-VQA	Beam size	Acc. on Val.
1	M4C [2]††	6N	2	-	✓	1	43.80
2	SA-M4C (ours)	2N→4S	2	Normal	✓	1	**45.10**
3	SA-M4C Rev	2N→4S	2	Reversed	✓	1	41.08
4	SA-M4C Rand	2N→4S	2	Randomized	✓	1	42.10

Performance on questions that involve spatial reasoning: Additionally, similar to our analysis in the main manuscript, we specifically look at the performance of questions that involve spatial reasoning. On this subset D_{spa} (∼14% of the dataset), the performance drops by 4.1% when the spatial graph is reversed (SA-M4C Rev), and drops by 2.6% when the spatial graph is randomly perturbed (SA-M4C Rand). Importantly, on $D_{\text{spa+ocr}}$ which consist of questions that require spatial reasoning and have a majority answer encoded in the OCR tokens, the performance drops drastically by 10% for SA-M4C Rev and 6.6% for SA-M4C Rand.

Visual Grounding: As a proxy to analyze visual grounding of our model, we look at instances in which models predict the answer using the list of OCR tokens without relying on the vocabulary. Our model (SA-M4C) picks an answer from the list of OCR tokens for 368/701 questions from the D_{spa} subset, and achieves 52.85% accuracy. In contrast, the SA-M4C Rev and SA-M4C Rand models

achieve 39.47% and 42.43% accuracy respectively. Similarly, on $D_{\text{spa+ocr}}$ SA-M4C achieves an accuracy of 67.95%, whereas SA-M4C Rev and SA-M4C Rand achieve 56.54% and 52.46% respectively.

In our model, each of the attention heads specializes in encoding a different spatial context. Consequently, we observe that reversing or randomly changing the spatial context for these heads by deliberate perturbations to the spatial graph has a notable affect on performance.

E Additional Experiments

ST-VQA Weakly Contextualized Task: We train SA-M4C with 30k vocabulary and achieve 49.7% ANLS accuracy beating the previous SoTA by 18.68% on the Weakly Contextualized Task of ST-VQA.

Adding fully connected heads in the spatial layer: We experimented with a model that extends the 12-head spatially-aware layer by adding 6 fully-connected heads that model all spatial relations while keeping the number of parameters comparable to the proposed approach. The performance drops from 43.8% to 43.41%.

F Qualitative Samples

Fig. 1: Qualitative Examples: The figure shows the output of M4C and our method on several image-question pairs. *Bold and italics text* denote words chosen from OCR tokens, otherwise it was chosen from the vocabulary. The VQA score for each prediction is mentioned inside parenthesis.

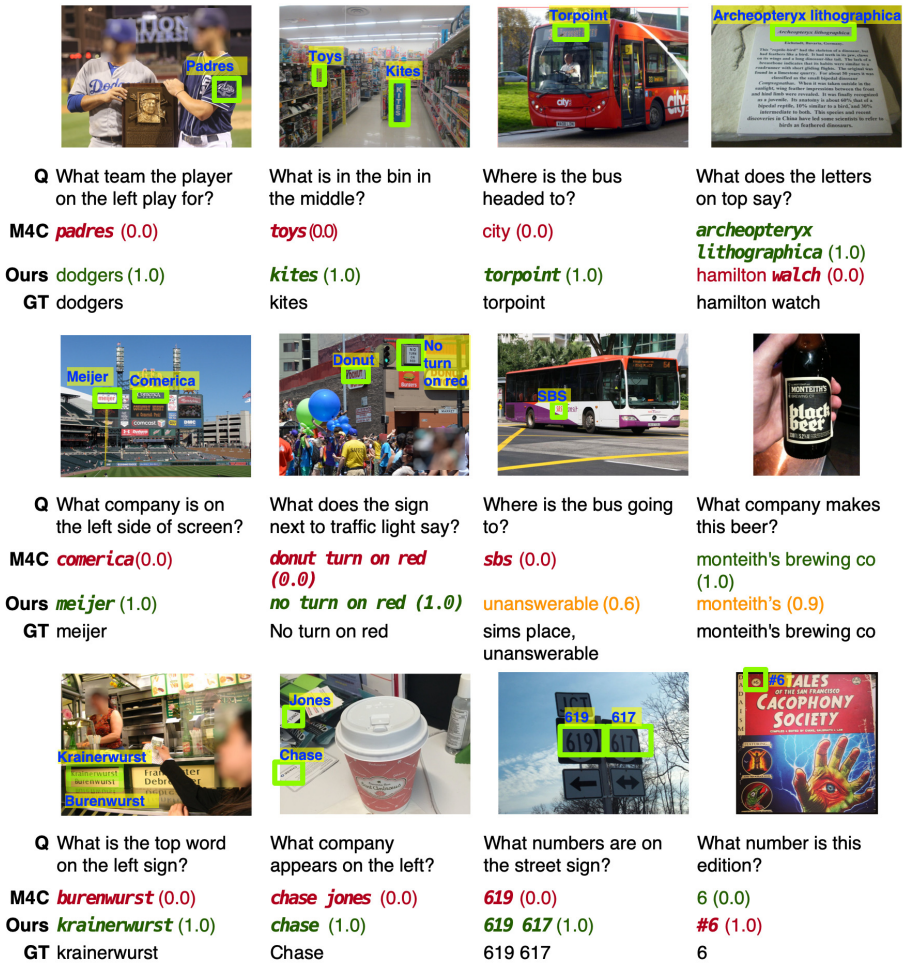

Fig. 2: Qualitative Examples: The figure shows the output of M4C and our method on several image-question pairs. *Bold and italics text* denote words chosen from OCR tokens, otherwise it was chosen from the vocabulary. The VQA score for each prediction is mentioned with parenthesis.

Fig. 3: Qualitative Examples: The figure shows the output of M4C and our method on several image-question pairs. *Bold and italics text* denote words chosen from OCR tokens, otherwise it was chosen from the vocabulary. The VQA score for each prediction is mentioned with parenthesis.

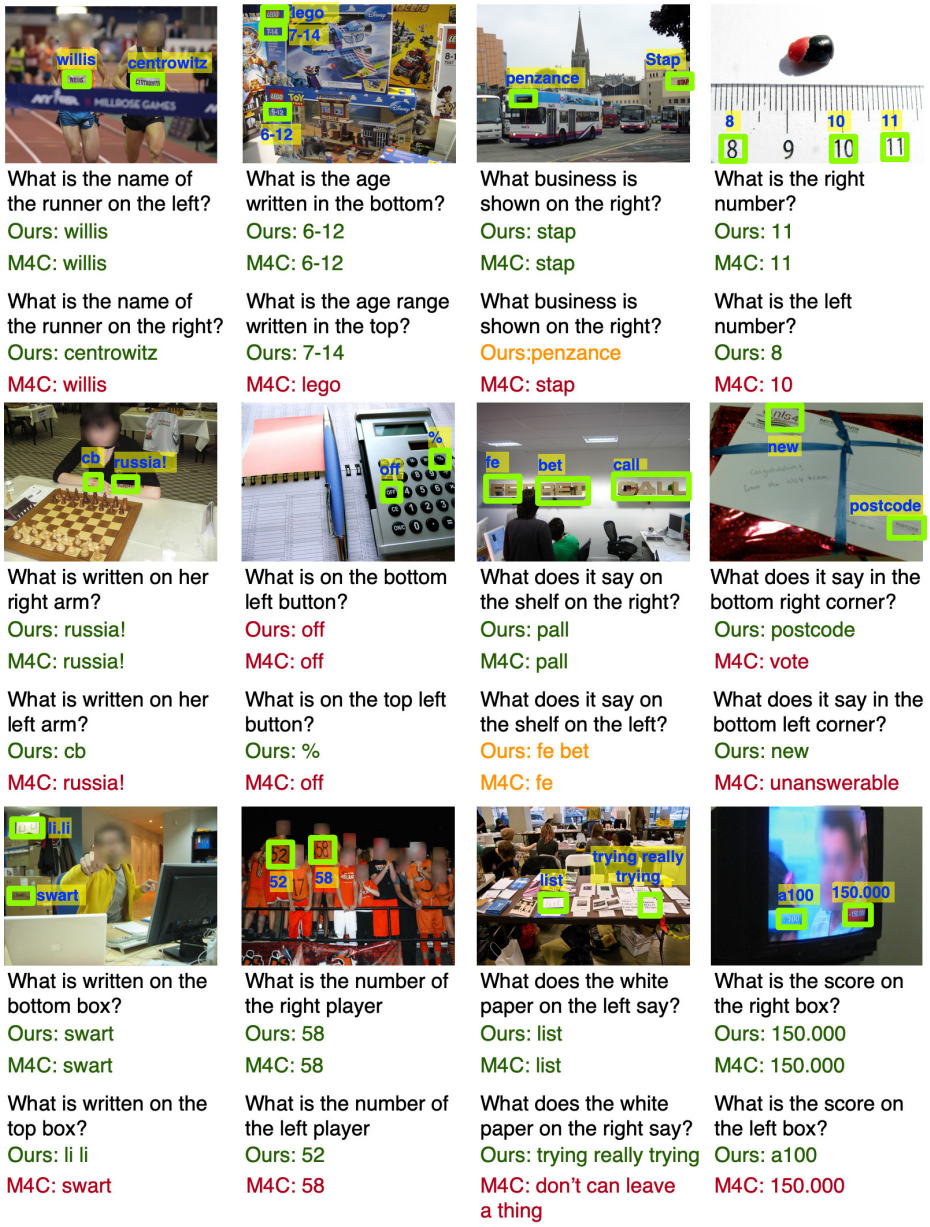

Fig. 4: The figure shows examples where we flipped the spatial relation in the original question to see whether the models change their answers. We observe that our spatially aware multimodal transformer correctly reasons about the spatial relationships mentioned in the question and predict the answer more accurately than M4C. Green text denote correct predictions. Red text denote incorrect predictions while orange text denote partially correct answers.

References

1. Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4291–4301, 2019. 1
2. Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. *arXiv preprint arXiv:1911.06258*, 2019. 2
3. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 1
4. 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. 1
5. Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8309–8318, 2019. 1