12-in-1: Multi-Task Vision and Language Representation Learning

8. Supplementary

In this section, we first show the full details of the cleaned dataset in Sec. 8.1. We further discuss the modifications in pretraining, show our multi-task model architecture and describe the implementation details in Sec. 8.2, Sec. 8.3 and Sec. 8.4 respectively. The rest of the section provides extensive experimental results to fully analyze our proposed model.

8.1. Datasets

Table 7 shows the number of images in the train+val and test sets before and after cleaning. Our cleaning process removes 13.02% of the total number of images on average. It is important to note that here we show the number of images per dataset and not the number of actual training samples. Different tasks have different number of training samples for each image. For details on training samples please refer to Table 8. We collect the union of all dataset test sets and remove any occurrence of these images from all training and validation sets; in this way we arrive at the *Clean* training and validation sets. With this strategy, the test sets of the original datasets are not modified in any way.

	Train+Val	Test	Cleaned Train+Val	% Removed
[A] VQA2.0 [15]	123,287	81,434	98,861	19.81
[B] VG QA [23]	108,249	-	92,147	14.87
[C] GQA [17]	82,374	2,987	69,868	15.18
[D] COCO Retrieval [7]	118,287	5,000	99,435	15.93
[E] Flickr30k Retrieval [41]	30,014	1,000	29,077	3.12
[F] RefCOCO [20]	18,494	1,500	14,481	21.69
[F] RefCOCO+ [20]	18,492	1,500	14,479	21.70
[H] RefCOCOG [35]	23,199	2,600	17,903	22.82
[I] Visual 7W [63]	17,953	7,780	16,415	8.56
[J] GuessWhat [13]	56,638	9,899	51,291	9.44
[K] SNLI-VE [56]	30,783	1,000	29,808	3.16
[L] NLVR^2 [51]	95,522	8,056	95,522	0
Average	-	-	-	13.02

Table 7: Number of images in the train+val and test sets before and after cleaning. We use the training part of the cleaned dataset in the multi-task experiments. Note that this is not the number of training samples but the number of images in the dataset.

8.2. Improvements over ViLBERT Pretraining

In this section, we discuss in detail the modification we made to the base ViLBERT pretraining approach.

Masked prediction with mislaigned pairs. In the original ViLBERT pretraining procedure, the model observes an image and caption as inputs. The caption is either obtained from the paired caption (with $p = 0.5$) or a randomly sampled misaligned caption from the dataset. The *multi-modal alignment prediction* task, which predicts whether the image and caption are aligned, is crucial for image retrieval tasks [26, 32, 52]. Recent work [50] has questioned the necessity of the *multi-modal alignment prediction* task and observed better performance on non-image retrieval tasks without this pretraining objective. Similar observations are also found in the natural language understanding tasks [19, 24, 30, 57]. Digging further into this, we find that both the alignment and prediction tasks are typically done together. For misaligned image-caption pairs, this amounts to forcing the model to predict missing image or text regions based on incorrect paired data! We find the model will learn worse context representations in this setup. Instead of removing the *multi-modal alignment prediction* task, we only perform the *mask multi-modal modelling* task on **aligned image-caption pairs**. This will effectively remove the noise introduced by negative samples.

Masking overlapping regions. Different from words embedding in the caption, visual feature embeddings (extracted from a pretrained Faster-RCNN [44]) have a lot of repetitions due to overlapped image regions. To avoid visual clue leakage from the visual embedding of other elements, VL-BERT [50] sets the pixels laid in the masked RoI to zeros before applying Faster R-CNN. However, overlapped image patches with boundary information may still leak the visual clues for the masked RoI. We mask the overlapped image regions in a more aggressive manner – any visual embedding that overlaps a masked region by 40% IOU or more is also masked. We observe significant improvements over the ViLBERT model as shown in Table 9 when comparing column ViLBERT with Ours$_{\mathtt{ST}}$.

8.3. Model Architecture

Fig. 5 shows the architecture of the our model for V&L multi-task learning, which is described in Sec. 3.2. We use ViLBERT as our base model shared across different tasks. For the task-specific heads, our model jointly train with four different task group – Vocab-Based VQA; Image Retrieval, Refer Expression and Multimodal Verification.

8.4. Implementation Details

Image features are extracted from a ResNeXT-152 Faster-RCNN model trained on Visual Genome(VG) with attribute loss. Our model is first initialized from pretrained BERT weights [14]. Our models are trained using AdamW optimizer [31] with a linear warmup and linear decay learning rate scheduler. We train our multi-task model for 40K total iterations (same as the number of iterations for the VG QA single task) on 8 NVIDIA V100 GPUs for 5 days. We use AdamW optimizer and a warmup linear schedule. Hyperparameters like learning rate and batch sizes used for

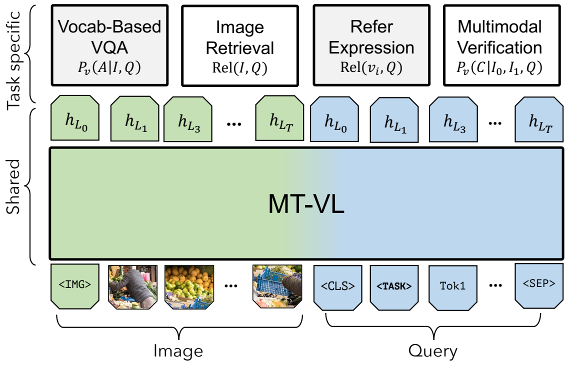

Figure 5: Architecture of the our model for V&L multi-task learning. We augment the input query with a task token to learn the task-aware feature embedding.

	Samples				Hyperparams	
	Full Train	Cleaned Train	Test	Metric	BS	LR
[A] VQA2.0 [15]	655,111	542,104	447,793	VQA Accuracy	128	4e-5
[B] VG QA [23]	1,437,931	1,294,255	5,000	VQA Accuracy	128	4e-5
[C] GQA [17]	1,072,062	962,928	12,578	VQA Accuracy	128	4e-5
[D] IR COCO [7]	566,747	487,600	1,000	Recall @ 1, 5, 10	128	2e-5
[E] IR Flickr30k [41]	145,000	140,485	1,000	Recall @ 1, 5, 10	128	2e-5
[F] RefCOCO [20]	120,624	96,221	10,752	Accuracy	256	2e-5
[F] RefCOCO+ [20]	120,191	95,852	10,615	Accuracy	256	2e-5
[H] RefCOCOG [35]	80,512	65,514	9,602	Accuracy	256	2e-5
[I] Visual 7W [63]	93,813	93,813	57,265	Accuracy	256	2e-5
[J] GuessWhat [13]	113,221	100,398	23,785	Accuracy	64	2e-5
[K] NLVR^2 [51]	86,373	86,373	6,967	Accuracy	64	2e-5
[L] SNLI-VE [56]	529,527	512,396	17,901	Accuracy	256	2e-5
Total	5,021,112	4,477,939	604,258	-	-	-

Table 8: Training details including sample sizes, testing metric and hyperparameters for single task and multi-task training.

each task are listed in Table 8. We also report the number of training samples used in various settings in our experiments.

8.5. Multi-Task Training

To further illustrate the multi-task training process, in Fig. 7 we show the training curves for single-task *vs.* multi-task for all the 12 tasks in our setup. Green lines show single-task training and blue lines show multi-task training. Since we train the model with maximum iterations across different datasets for multi-task training, for some smaller datasets (*e.g.* RefCOCO, Visual7W *etc.*), the number of iterations for single task is much smaller compared to the multi-task setting. By comparing the training curves of single-tasks and multi-tasks, we can see that most of the tasks have similar training curves. However, the tasks in the vocab-based VQA group benefit from the multi-task training with faster convergence within first 10000 iterations.

Concept drift of smaller datasets. In Fig. 6 (left), we plot the val accuracy of our AT model on RefCOCO+ to show the concept drift of smaller datasets during MT training. Even with sparse updates (stop mode), we observe sharp drops (dips before go mode is reactivated) on RefCOCO+.

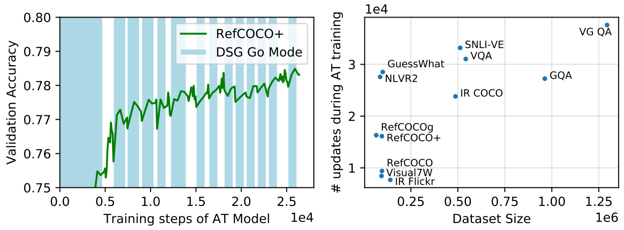

Figure 6: Left: Val acc. of our AT model on RefCOCO+. Right: Dataset size vs. number of updates during stop-and-go training.

Relationship between dataset size and `go` mode duration. The dataset size gap can be significant – up to 16:1 for VG QA vs ReferCOCOg. To illustrate how dataset size affects our dynamic stop-and-go training regime, we plot dataset size vs active training iterations in Fig. 6 (right). Among datasets with a similar size, we see significant differences in active training time. This shows that dynamic stop-and-go addresses issues of dataset difficulties rather than just size. However, there is a general trend that larger datasets do tend to stay in the active `go` mode longer.

Full per task accuracy for `AT` without G4 model. In Table 2, we observed from the representative task analysis that G4 tends to have a negatively effect other group during joint training. In Table 12, we further show full per task accuracy for `AT` without G4 model and different ablations. We can see that $\texttt{AT}_{\text{w/o G4}}$ outperforms `AT` 0.48% on MT scores, which verifies G4 tends to have negatively effect even on the finetuned model. How to remove the negative interactions between different tasks is left to future study.

8.6. Comparison with other SOTA

Table 9 shows the detailed comparison of Ours$_{\texttt{ST}}$ (also shown in Table 2, line 1) and Ours$_{\texttt{AT->ST}}$ (also shown in Table 2, line 8) with the recent SOTA approaches, inlcuding ViLBERT [32], Unicoder-VL [26], VisualBERT [27], LXMERT [52] and UNITER [8]. Most of the recent proposed methods follows the pretrain-then-finetune scheme, usually pretraining on out-of-domain data or in-domain data. The out-of-domain data contains Conceptual Caption Dataset (CC) [46] and SBU dataset [39] while in-domain data contains COCO [7] and Visual Genome [22]. Pretraining on the in-domain datasets usually leads to better downstream performance, since there is less domain transfer from pretraining to finetuning. Similar to ViLBERT, we pretrain our model on CC, which is different from VL-BERT (CC + Wiki Corpus), VisualBERT (CC + COCO), LXMERT (COCO + VG) and UNITER (CC + SUB + COCO + VG). We achieve comparable performance with less pretrained data.

8.7. Full Breakdown of Ablation Study

Table 10 shows the full breakdown of Table 6 and Fig. 2 per task in the main paper. RC refers to Retrieval COCO and RF refers to Retrieval Flickr30k. VQA and GQA

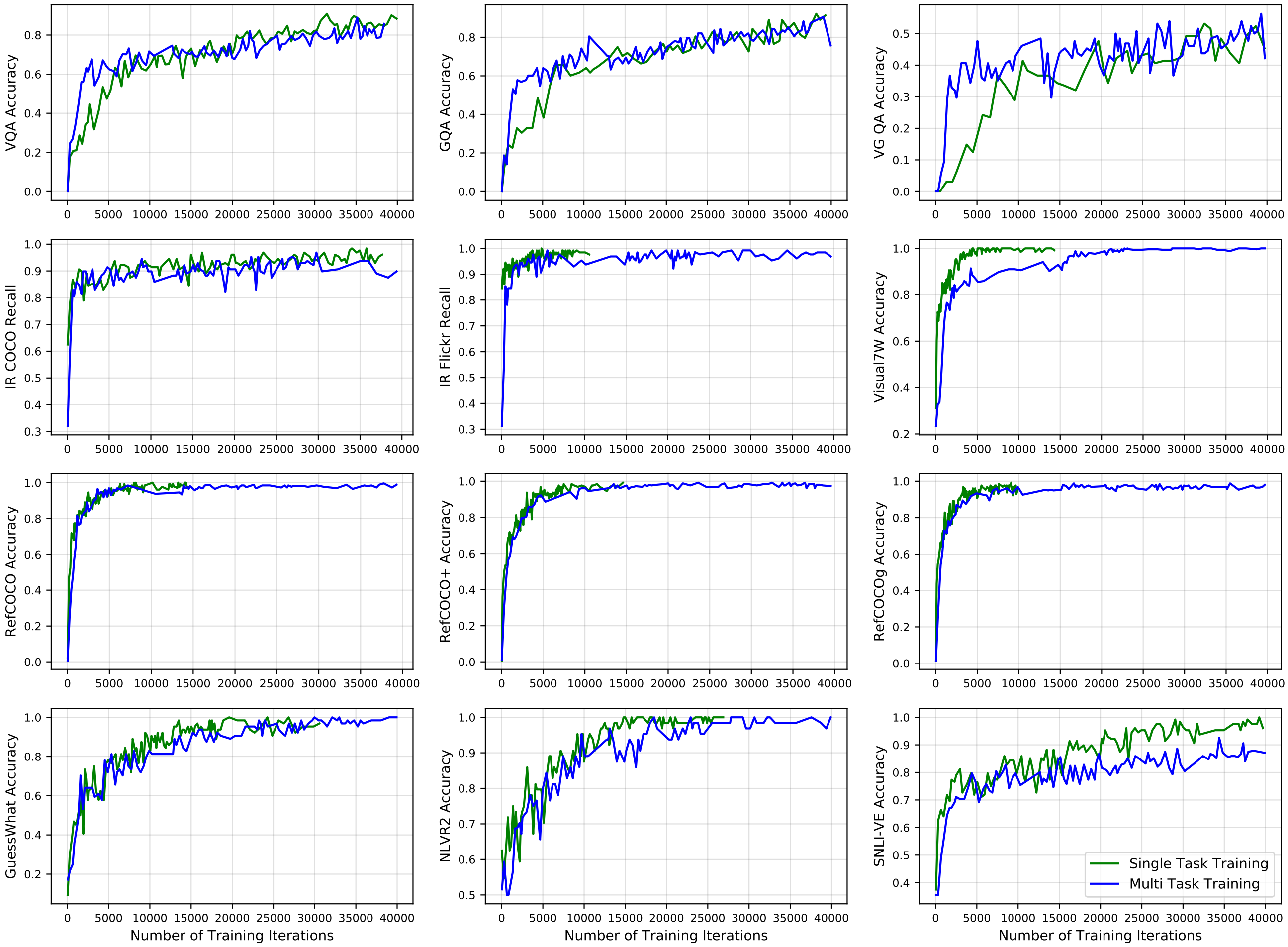

Figure 7: Training curves on *train* set for Ours$_{\texttt{ST}}$ (Table 2 `Row 2`) vs Ours$_{\texttt{AT}}$ (Table 2 `Row 4`) models for all the 12 tasks in our experiments. Green lines show single-task training(Ours$_{\texttt{ST}}$) and blue lines show multi-task training(Ours$_{\texttt{AT}}$). Note that all these training are with the *Clean V&L* setup. We can observe that for some of the tasks the training for Ours$_{\texttt{ST}}$ are shorter as they have fewer number of iterations when trained alone. Please refer to Sec. 8.5 for more details.

are evaluated on `test-dev` splits. Retrieval COCO and Flickr30k are evaluated on their respective 1K test split. NLVR^2 is evaluated on `testP` split. All other datasets are evaluated on their respective test splits. Table 11 shows the full scores for each task for different `DSG` iteration gap (Δ). Table 12 shows the detailed per task scores for `AT`$_{\text{w/o G4}}$ model and different ablations for it. We compare with full `AT` model as well.

8.8. Multi-task visual grounding consistency

In Sec. 5, we propose the multi-task visual grounding consistency. Here, we explain the proposed metric in more detail. Given N images with RefCOCO/+ refer expression and VQA questions, we want to test if multi-task models exhibit more consistent visual groundings than independent task-specific models. For each image I_i, there are associated VQA question $\{q^{(i)}\}$ and referring expression $\{r^{(i)}\}$. To measure the overlap in visual concepts between a question $q_j^{(i)}$ and reference $r_k^{(i)}$, we count the the number of overlapped noun / adj as $d(q_j^{(i)}, r_k^{(i)})$, the multi-task visual grounding consistency can be calculated as:

$$\texttt{MT-VGC} = \frac{\sum_{k=0}^{N}|\sum_j \sum_k d(q_j^{(i)}, r_k^{(i)})\mathbb{1}_{\{y(q_j^{(i)})=1\&y(r_k^{(i)})=1\}}|}{\sum_{i=0}^{N}|\sum_j \sum_k d(q_j^{(i)}, r_k^{(i)})\mathbb{1}|} \qquad (5)$$

where $y(q_k^{(i)}) = 1$ means the model correctly answer the question $q_k^{(i)}$ based on VQA accuracy metric and $y(r_k^{(i)}) = 1$ means the model correctly locate the image regions (IoU > 0.5) given the reference $r_k^{(i)}$.

8.9. Qualitative Results

Fig. 8 shows more qualitative examples of our single model Our$_{\texttt{AT}}$ on different vision and language tasks and Fig. 9 shows some failure cases. The examples in Fig. 8 show that the `AT` model works well for these wide range of

Tasks		SOTA	ViLBERT	VLBERT	Unicoder-VL	VisualBERT	LXMERT	UNITER		Ours$_{\texttt{ST}}$	Ours$_{\texttt{AT->ST}}$
								BASE	LARGE		
	Pretraining Data		CC	CC + Wiki Corpus	CC	CC + COCO	COCO + VG	CC+SUB+COCO+VG		CC	CC
VQA	test-dev	70.63	70.55	70.50	-	70.80	72.42	72.27	**73.24**	71.82	73.15
VG QA	val	-	-	-	-	-	-	-	-	34.38	**36.64**
GQA	test-dev	-	-	-	-	-	60.00	-	-	58.19	**60.65**
IR COCO	R1	61.60	-	-	**68.50**	-	-	-	-	65.28	68.00
	R5	89.6	-	-	**92.70**	-	-	-	-	91.02	92.38
	R10	95.2	-	-	**96.90**	-	-	-	-	96.18	96.52
IR Flickr	R1	48.60	58.20	-	68.30	-	-	71.50	**73.66**	61.14	67.90
	R5	77.70	84.90	-	90.30	-	-	91.16	**93.06**	87.16	89.60
	R10	85.20	91.52	-	94.60	-	-	95.20	**95.98**	92.30	94.18
Visual 7W	test	72.53	-	-	-	-	-	-	-	80.51	**83.35**
Ref-COCO	test	77.12	-	-	-	-	-	80.48	80.88	78.63	**81.20**
Ref-COCO+	test	67.17	70.93	69.47	-	-	-	73.26	73.73	71.11	**74.22**
Ref-COCOg	test	69.46	-	-	-	-	-	74.51	75.77	72.24	**76.35**
GuessWhat	test	61.30	-	-	-	-	-	-	-	62.81	**65.69**
NLVR^2	test-P	53.50	-	-	-	67.00	74.50	77.87	**79.50**	74.25	78.87
SNLI-VE	test	71.16	-	-	-	-	-	78.02	**78.98**	76.72	76.95

Table 9: Comparison of Ours$_{\texttt{ST}}$ (Table 2 `Row 1`) and Ours$_{\texttt{AT->ST}}$ (Table 2 `Row 8`) models on full dataset with other SOTA methods. Results for RefCOCO and RefCOCO+ are reported on the full test split (testA + testB). Refer to Sec 8.6 for more details.

	VQA	VG QA	GQA	Mean G1	RC R@1	RC R@5	RC R@10	RF R@1	RF R@5	RF R@10	Mean G2 (R1)	RefCOCO	RefCOCO+	RefCOCOG	Visual 7W	GuessWhat	Mean G3	NLVR^2	SNLI-VE	Mean G4	MT Score
token per dataset	**72.57**	**36.36**	60.12	**56.35**	**63.70**	**90.84**	**96.16**	**63.52**	**87.48**	**93.16**	**63.61**	80.58	73.25	**75.96**	82.75	**65.04**	**75.52**	**78.44**	76.78	**77.61**	**69.08**
token per head	72.11	35.84	59.91	55.95	60.66	88.96	94.86	62.30	86.20	92.00	61.48	**80.67**	73.10	75.82	**82.92**	64.24	75.35	77.65	**77.08**	77.37	68.52
w/o task token	72.00	35.09	59.92	55.67	63.16	90.48	95.44	61.94	86.96	92.88	62.55	80.32	73.04	75.94	82.72	64.89	75.38	76.99	76.46	76.73	68.53
w/o `DSG`	71.99	35.59	58.93	55.50	62.54	90.08	95.42	63.30	86.98	92.86	62.92	79.99	73.09	75.94	82.68	64.52	75.24	77.37	76.31	76.84	68.52
w/ curriculum	70.59	35.54	57.91	54.68	61.14	89.74	95.04	61.28	86.58	92.56	61.21	80.11	73.35	75.62	82.38	64.51	75.19	77.20	76.19	76.69	67.98
w/ anti-curriculum	71.53	35.54	**60.39**	55.82	61.04	88.78	94.96	58.12	84.66	90.84	59.58	78.99	71.34	74.24	80.80	63.08	73.69	76.14	75.74	75.94	67.24
vanilla multitask	70.39	33.31	58.57	54.09	61.50	89.72	95.42	61.40	87.04	92.74	61.45	80.42	**73.51**	75.53	82.48	64.50	75.28	77.09	76.34	76.71	67.92
w/o CC pretraining	70.23	33.49	58.41	54.04	57.92	87.60	93.96	56.72	83.20	90.68	57.32	77.93	69.60	72.21	78.99	61.67	72.08	73.63	75.92	74.77	65.56

Table 10: Full per task accuracy for the different ablation studies (summarized in Table 6). RC is Retrieval COCO and RF is Retrieval Flickr30k. Mean of G2 is taken over the Recall@1 scores. We can see that with task token per dataset and `DSG` achieve the best performance.

tasks consistently. It can perform well in both short as well as long reasoning questions, image retrieval, pointing tasks, referring expressions and multi-modal validation. Failure cases mostly occur when the model encounters `counting` questions or difficult referring expressions and phrases for fine grained recognition.

8.10. Attention Visualizations

In this section we examine the visual groundings learned by the techniques we presented in Sec. 8.2. We verify this by visualizing the attentions of our pretrained model, which is trained on the Conceptual Caption dataset. Given a test image, and corresponding caption "The boy and his mom pet the black and white sheep", we feed the image-caption pair as input and take the image to question co-attention for visualization. For each image patch, we use the most attended word to represent its semantic meaning, and show the patches corresponding to the visual words ('boy', 'mom', 'pet', 'white', 'sheep'). Fig. 10 shows the correspondence between attended regions and underlined words. We can see that the pretrained model learns meaningful visual grounding for the concept 'boy', 'sheep', 'white' and 'pet'.

To visualize the attention for our multi-task trained model (Ours$_{\texttt{AT}}$), we use BertVis[2] to visualization the attention distribution on the sentence to sentence self-attention $S{\rightarrow}S$, sentence to image co-attention $S{\rightarrow}I$, image to sentence co-attention $I{\rightarrow}S$ and image to image self attention $I{\rightarrow}I$. Fig. 11 shows an example of the sentence to sentence attention for all layers and all heads (middle) and a specific layer and head (right). We can see that our model learns the previous words attention pattern, bag of words at-

[2] https://github.com/jessevig/bertviz

	VQA	VG QA	GQA	Mean G1	RC R@1	RC R@5	RC R@10	RF R@1	RF R@5	RF R@10	Mean G2 (R1)	RefCOCO	RefCOCO+	RefCOCOG	Visual 7W	GuessWhat	Mean G3	NLVR^2	SNLI-VE	Mean G4	MT Score
DSG Δ1	71.99	35.59	58.93	55.50	62.54	90.08	95.42	63.30	86.98	92.86	62.92	79.99	73.09	75.94	82.68	64.52	75.24	77.37	76.31	76.84	68.52
DSG Δ4	72.57	36.36	**60.12**	**56.35**	63.70	90.84	**96.16**	63.52	87.48	**93.16**	63.61	80.58	73.25	**75.96**	82.75	65.04	75.52	**78.44**	**76.78**	**77.61**	69.08
DSG Δ8	72.61	**36.65**	59.69	56.32	**65.24**	90.86	96.02	**63.56**	87.60	93.08	**64.40**	80.32	**73.56**	75.88	**82.79**	**65.33**	**75.58**	77.43	76.75	77.09	**69.15**
DSG Δ16	**72.74**	35.34	59.70	55.93	64.78	**91.04**	95.86	62.36	**87.66**	92.92	63.57	**80.59**	73.17	75.88	82.61	64.79	75.41	78.18	76.66	77.42	68.90

Table 11: Full per task accuracy for Fig. 2 showing different Dynamic Stop-and-Go Iteration Gaps (Δ). Mean of G2 is taken over the Recall@1 scores.

	VQA	VG QA	GQA	Mean G1	RC R@1	RC R@5	RC R@10	RF R@1	RF R@5	RF R@10	Mean G2 (R1)	RefCOCO	RefCOCO+	RefCOCOG	Visual 7W	GuessWhat	Mean G3	MT Score
AT	72.57	36.36	60.12	56.35	63.70	90.84	96.16	63.52	87.48	93.16	63.61	80.58	73.25	**75.96**	82.75	65.04	75.52	56.15
AT$_{\text{w/o G4}}$	**72.68**	**36.74**	**62.09**	**57.17**	**64.88**	**91.36**	95.98	**64.62**	87.98	93.18	**64.75**	**80.76**	**73.60**	75.80	**83.03**	**65.41**	**75.72**	**56.63**
w/o task token	71.54	34.42	61.62	55.86	64.34	90.80	**96.18**	63.24	86.86	92.52	63.79	80.53	72.77	75.33	82.79	64.52	75.18	55.92
w/o DSG	71.70	34.15	59.82	55.22	63.20	90.70	96.04	63.44	**88.18**	**93.28**	63.32	80.64	72.86	75.81	82.56	64.76	75.32	55.74

Table 12: Full per task accuracy for AT$_{\text{w/o G4}}$ model and different ablations for AT$_{\text{w/o G4}}$. Mean of G2 is taken over the Recall@1 scores.

tention pattern (Layer 1, Head 1) and next words attention pattern (Layer 2, Head 0). This shows that the model is able to generate position-aware queries and keys to calculate the attentions. To get a sense of the difference of attention distribution across different tasks, Fig. 12 and Fig. 13 show the attention distribution on the examples of Fig. 3. We can see for different tasks, the model learns to use significant different sentence to sentence self-attention pattern.

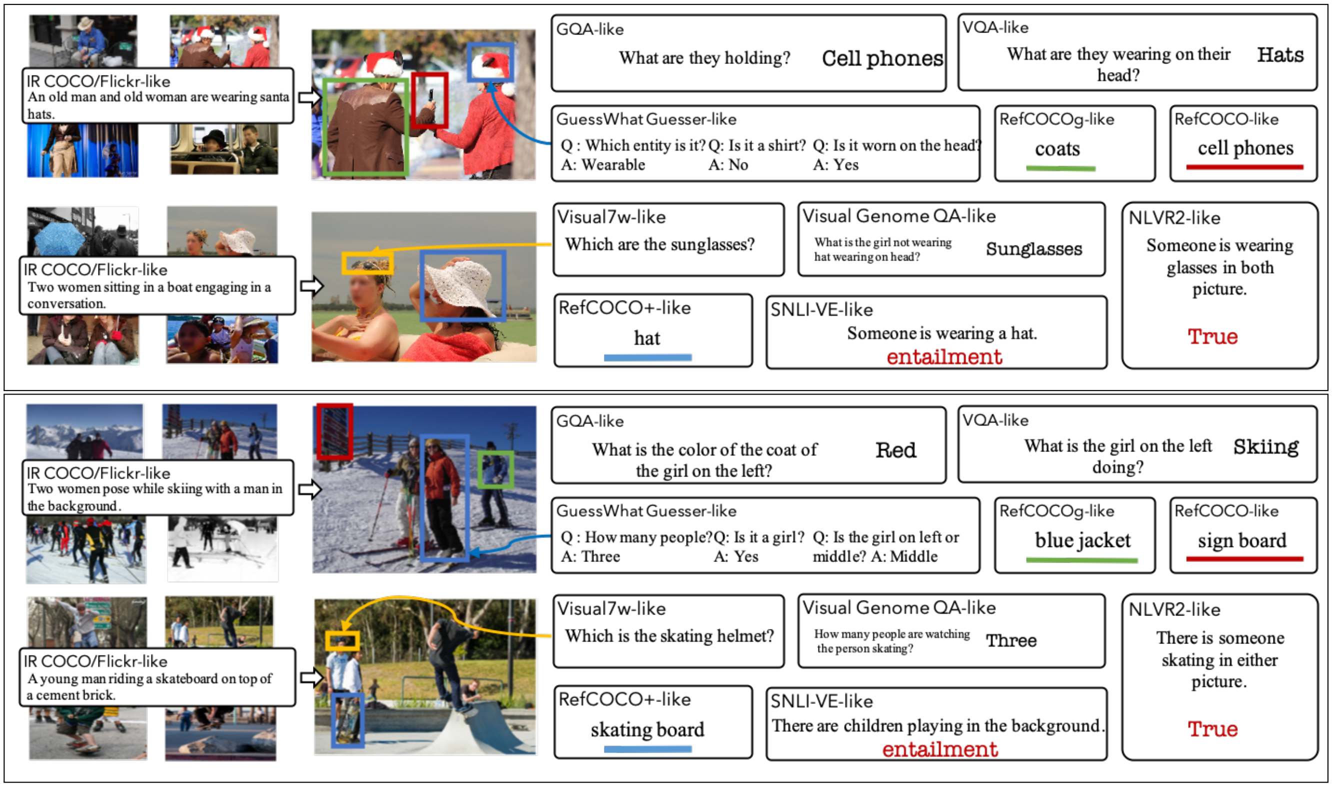
IR COCO/Flickr-like
An old man and old woman are wearing santa hats.
GQA-like
What are they holding? Cell phones
VQA-like
What are they wearing on their head? Hats
GuessWhat Guesser-like
Q : Which entity is it? A: Wearable
Q: Is it a shirt? A: No
Q: Is it worn on the head? A: Yes
RefCOCOg-like
coats
RefCOCO-like
cell phones
IR COCO/Flickr-like
Two women sitting in a boat engaging in a conversation.
Visual7w-like
Which are the sunglasses?
Visual Genome QA-like
What is the girl not wearing hat wearing on head? Sunglasses
NLVR2-like
Someone is wearing glasses in both picture.
True
RefCOCO+-like
hat
SNLI-VE-like
Someone is wearing a hat.
entailment
IR COCO/Flickr-like
Two women pose while skiing with a man in the background.
GQA-like
What is the color of the coat of the girl on the left? Red
VQA-like
What is the girl on the left doing? Skiing
GuessWhat Guesser-like
Q : How many people? A: Three
Q: Is it a girl? A: Yes
Q: Is the girl on left or middle? A: Middle
RefCOCOg-like
blue jacket
RefCOCO-like
sign board
IR COCO/Flickr-like
A young man riding a skateboard on top of a cement brick.
Visual7w-like
Which is the skating helmet?
Visual Genome QA-like
How many people are watching the person skating? Three
NLVR2-like
There is someone skating in either picture.
True
RefCOCO+-like
skating board
SNLI-VE-like
There are children playing in the background.
entailment

Figure 8: Our single multi-task model can solve multiple task consistently and correctly. Additional qualitative examples of our single model Our$_{\texttt{AT}}$ on multitude of V&L tasks: caption and image retrieval, question answering, grounding phrases, guessing image regions based on a dialog, verifying facts about a pair of images, natural language inferences from an image, etc. Here we show outputs of our model for a variety of inputs (that mimic tasks from the 12 datasets it has been trained on).

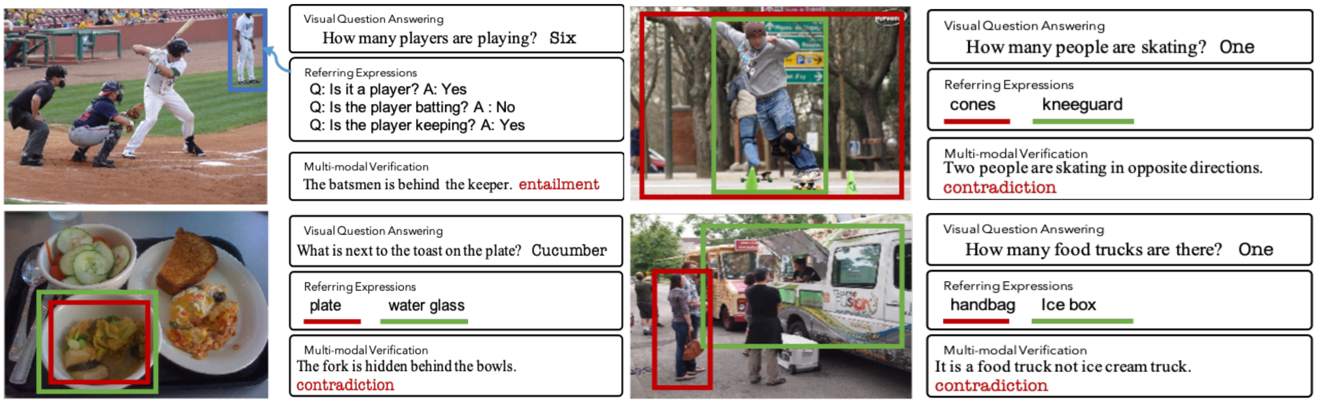
Visual Question Answering
How many players are playing? Six
Referring Expressions
Q: Is it a player? A: Yes
Q: Is the player batting? A : No
Q: Is the player keeping? A: Yes
Multi-modal Verification
The batsmen is behind the keeper. entailment
Visual Question Answering
How many people are skating? One
Referring Expressions
cones
kneeguard
Multi-modal Verification
Two people are skating in opposite directions.
contradiction
Visual Question Answering
What is next to the toast on the plate? Cucumber
Referring Expressions
plate
water glass
Multi-modal Verification
The fork is hidden behind the bowls.
contradiction
Visual Question Answering
How many food trucks are there? One
Referring Expressions
handbag
Ice box
Multi-modal Verification
It is a food truck not ice cream truck.
contradiction

Figure 9: Failure cases of our single `AT` model on multitude of V&L tasks. Failure cases mostly occur when the model encounters `counting` questions or difficult referring expressions and phrases for fine grained recognition.

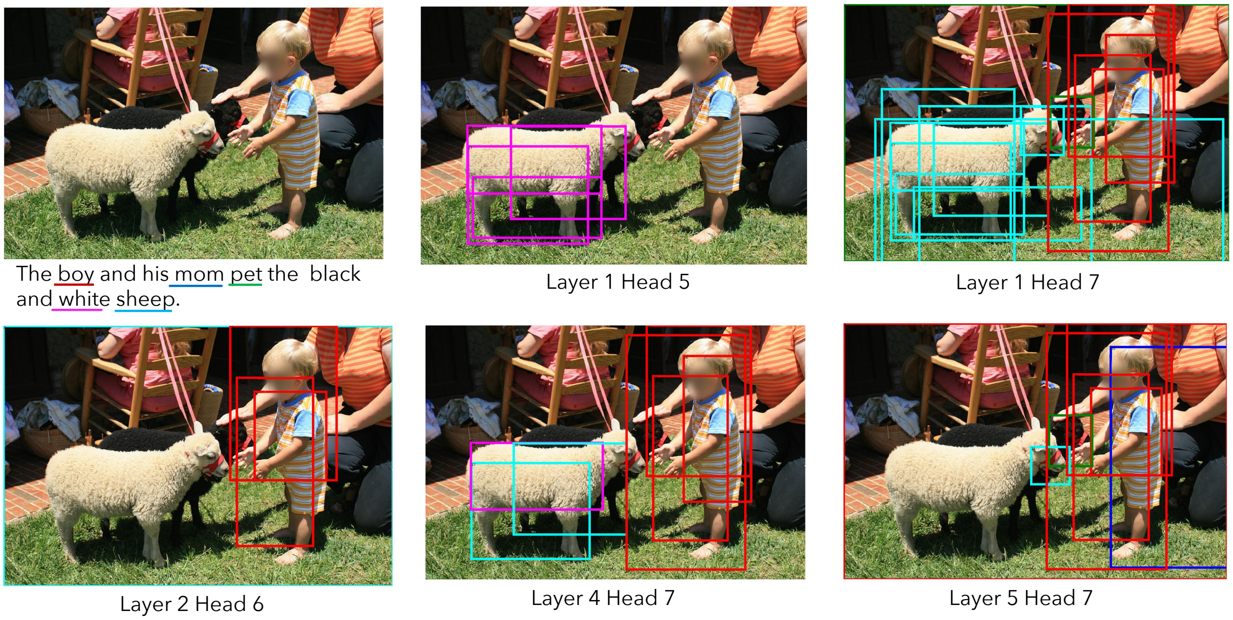

Figure 10: Visualizations of image to sentence attention for the pretrained model on conceptual caption dataset. Given the image to sentence co-attention, we use the most attended word to represent its semantic meaning, and show the patches corresponding to the visual words ('boy', 'mom', 'pet', 'white', 'sheep'). Different colors show a correspondence between attended regions and underlined words. We can see that the model learns meaningful concept through pretraining.

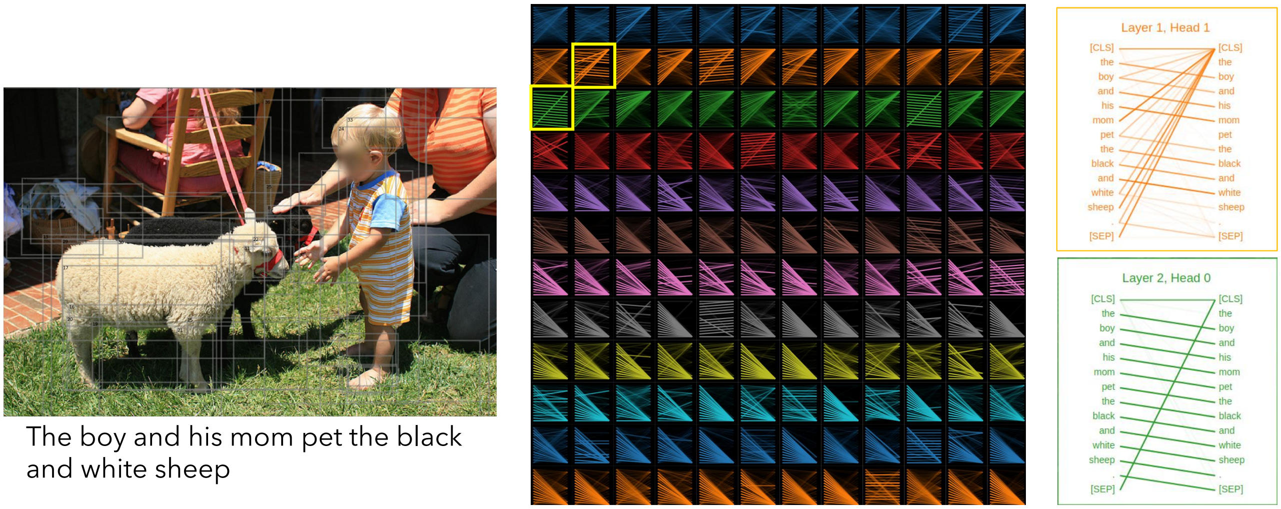

Figure 11: Visualizations of the attentions of the pretrained model on conceptual caption dataset using BertVis toolbox. From left to right: Image and associate caption, sentence to sentence self-attention for all layers and all heads, sentence to sentence self-attention for Layer 1 Head 1 and Layer 2 Head 0. Our model learns the previous words attention pattern, bag of words attention pattern and next words attention pattern.

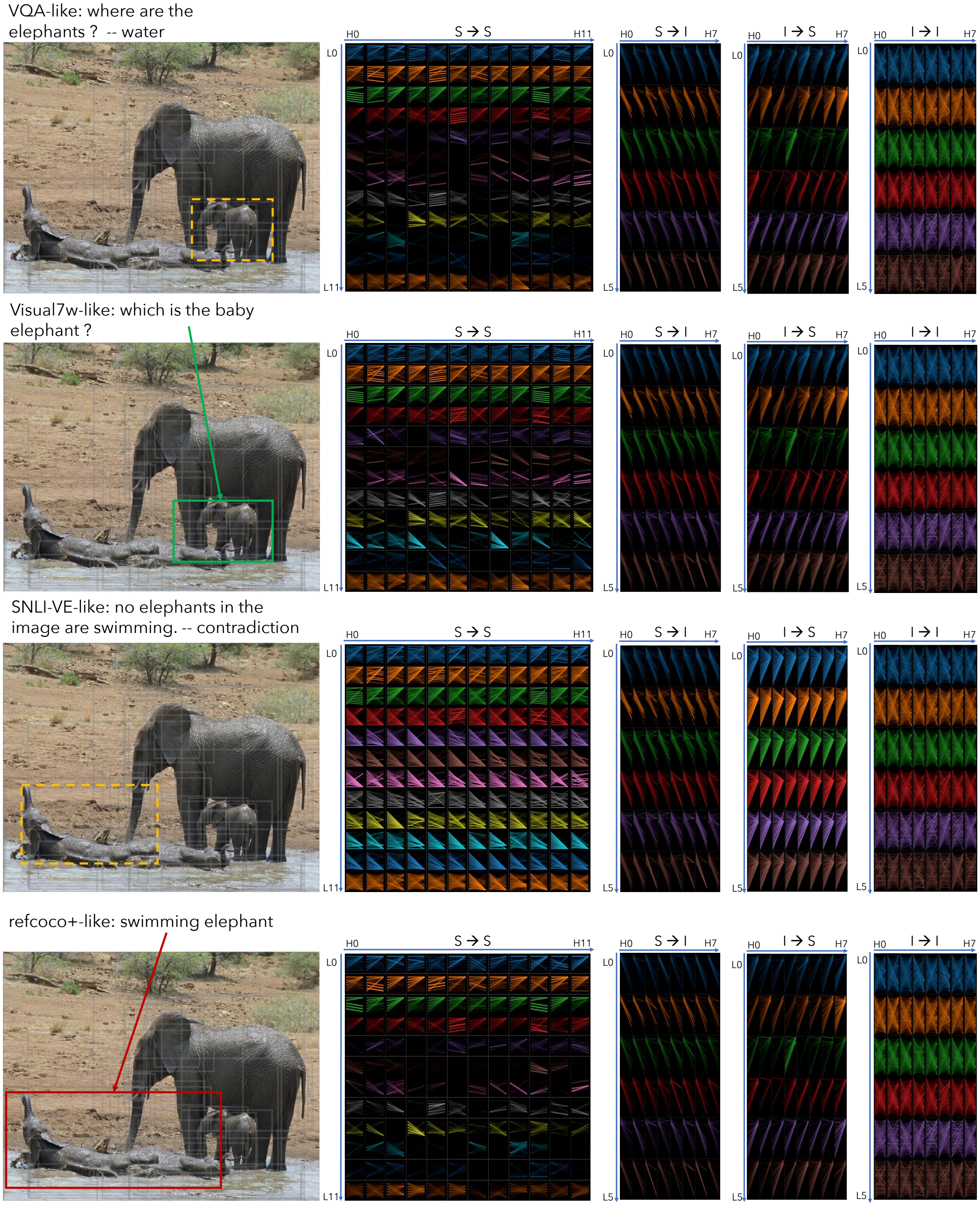

Figure 12: Visualizations of the attentions of Our$_{\mathtt{AT}}$ model using BertVis toolbox on each tasks. From left to right are image and associate sentence, sentence to sentence self-attention, sentence to image co-attention image to sentence co-attention image to image self-attention. Dashed orange bounding boxes in the image are the referring expression outputs regardless of tasks. The model learns to use significant different sentence to sentence self-attention pattern for different tasks.

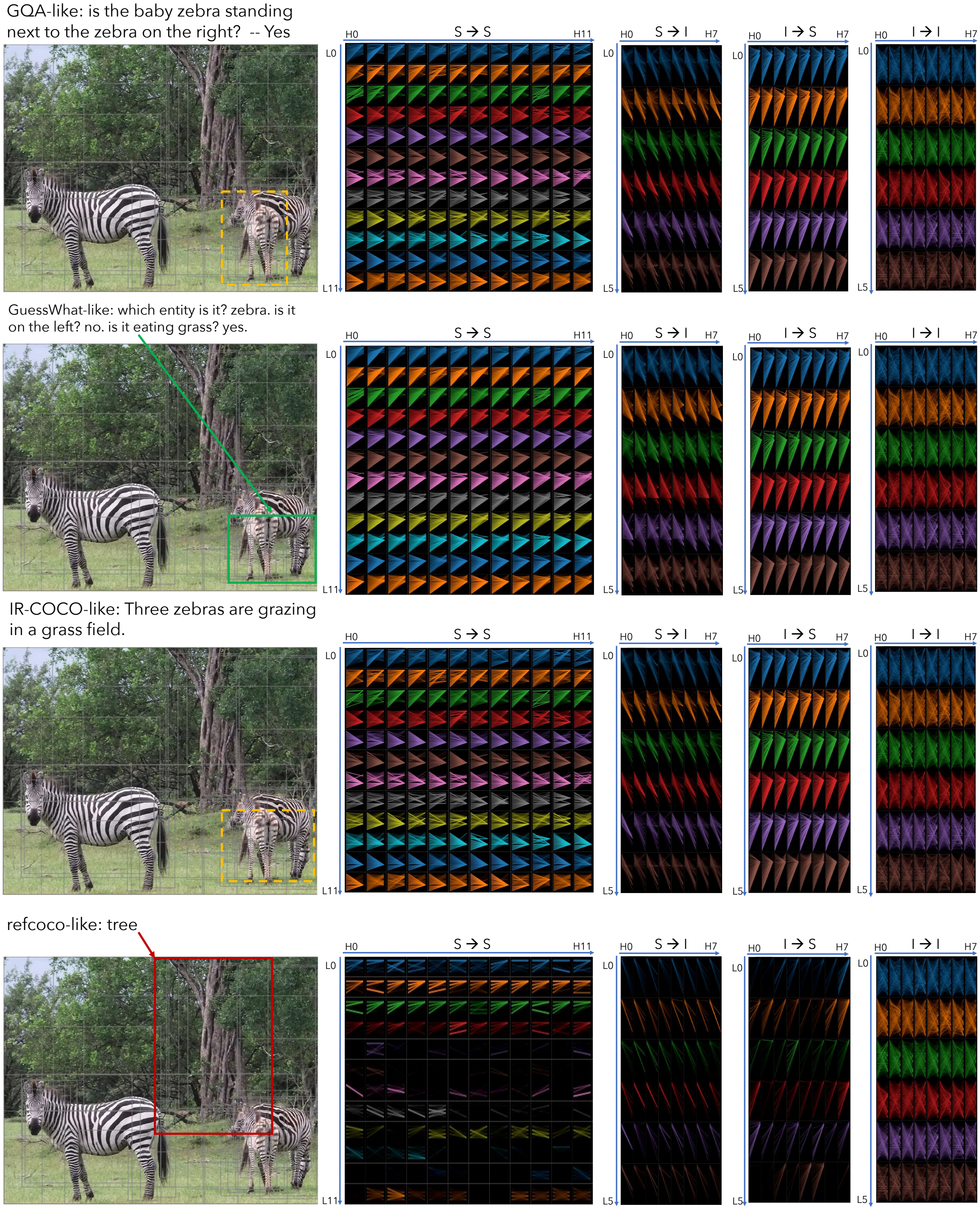

Figure 13: Visualizations of the attentions of Our$_{\texttt{AT}}$ model using BertVis toolbox on each tasks. From left to right are image and associate sentence, sentence to sentence self-attention, sentence to image co-attention image to sentence co-attention image to image self-attention. Dashed orange bounding boxes in the image are the referring expression outputs regardless of tasks. The model learns to use significant different sentence to sentence self-attention pattern for different tasks.