UniT: Multimodal Multitask Learning with a Unified Transformer
(Supplementary Material)

A. Hyper-parameters and details of UniT

We summarize the hyper-parameters in our UniT model in Table A.1. We also list the sampling probabilities of each dataset during joint training in Table A.2 under different experimental settings.

Unused parameters in the optimizer. Some parameters in our model (*e.g.* the task-specific output heads) are only used on a subset of tasks and datasets. During development, we first tried updating all parameters in the model during training even if some parameters were not used in the forward pass of a batch and their gradients remained zero. However, we empirically found that this strategy sometimes caused the training to diverge. On the other hand, the alternative strategy of skipping optimizer updates (including momentum accumulation) on unused parameters in a batch with zero gradients provides more stable training – however, in some cases, this alternative training strategy yields slightly lower scores (*e.g.* -0.2% lower accuracy on VQAv2).

When jointly training on COCO detection, VG detection, and VQAv2 with a shared decoder (Sec. 4.1 in the main paper), divergence happens if we update unused parameters in the optimizer, where the VQA accuracy stays around 25%. The divergence might be related to a high overall sampling probability on detection (0.667), such that the detection gradients dominate the model. We find that the alternative strategy (skipping unused parameters in optimizer) allows the model to converge properly in this case. Meanwhile, lowering sampling probabilities on detection datasets also avoids such divergence on VQA, but gives lower detection mAP than this alternative strategy.

Hyper-parameter	**Value**
image encoder hidden size	256
image encoder head number	8
image encoder intermediate size	2048
image encoder layer number	6
image encoder dropout	0.1
decoder hidden size	768
decoder head number	8
decoder intermediate size	2048
decoder layer number	6
decoder dropout	0.1
batch size	64
learning rate	5e-5
learning schedule	`warmup_cosine`
warmup iterations	2000
Adam β_1	0.9
Adam β_2	0.999

Table A.1: A list of hyper-parameters in UniT.

B. Multitask learning in UniT

In this work, we propose UniT – a multi-task joint model across several domains achieving comparable performance to per-task models with $8\times$ fewer parameters. As discussed in Sec. 2 in the main paper, our model is notably different from previous work in the pretrain-and-transfer paradigm – UniT is a joint and shared model instead of separately fine-tuned ones.

While per-task fine-tuning could be useful for single-task performance (and its results show that UniT can achieve competitive single-task performance), it is not ideal towards this multi-task goal, as one needs to save 8 separately fine-tuned models to handle all 8 tasks, leading to $8\times$ total parameters compared to a single shared UniT model.

In Table 3 in the main paper, our multi-task model (line 5) achieves better performance on VQAv2 and SNLI-VE but does not outperform separately-trained single-task models on pure vision or pure language tasks in line 1. We note that while multi-task learning sometimes benefits individual tasks, there is not much prior evidence on vision-and-language tasks helping pure vision tasks in a joint model via multi-task learning (instead of pretraining). In particular, no prior work to the best of our knowledge shows VQA, as compared to captioning, helps object detection via multi-task learning. Rather, better VQA accuracy often comes at sacrificing detection performance as detectors used in VQA are heavily specialized, *e.g.* the detector trained in BUTD [1] has relatively poor localization performance on COCO classes.[1] Meanwhile, we handle *both* detection and VQA with strong and comparable performance to prior work. Similarly, on vision-and-language and pure language tasks, we find that VisualBERT [3] has a noticeable drop on GLUE accuracy[2] over the original BERT, while our model solves vision-and-language tasks, GLUE as well as detection jointly with reasonable performance.

We emphasize that UniT handles all tasks in a shared model, where knowledge on object detection and language is not lost due to specializing to other tasks, in contrast to prior work on pretrain-and-transfer. We believe UniT's abil-

[1] on COCO classes: 15.2 mAP@IoU=0.5 and 5.0 mAP@IoU=0.5:0.95
[2] drop on QNLI, MNLI, QQP, SST-2: -2.76, -2.50, -0.70, -2.06

#	Experimental setting	COCO det.	VG det.	VQAv2	SNLI-VE	QNLI	MNLI-mm	QQP	SST-2
1	detection + VQA (Sec. 4.1)	0.33	0.33	0.33	–	–	–	–	–
2	all 8 tasks (Sec. 4.2)	0.20	0.07	0.26	0.12	0.10	0.10	0.10	0.05
3	ablation study (Sec. 4.2)	0.30	–	–	0.50	–	0.20	–	–

Table A.2: Sampling probabilities of each dataset for joint training under different experimental settings.

ity to jointly solve different tasks across domains is a critical step towards general intelligence.

Also in our experiments, we show that UniT can be applied over a diverse set of tasks through a shared model, even if some of them are usually considered unrelated (such as object detection in vision and sentiment analysis in language). This confirms that task compatibility is not a strict requirement for UniT to learn a joint shared model. On the other hand, we also find that some tasks are more compatible than others for joint training. There are both benefits from joint multi-task learning (because they can share supervision) and competitions between tasks (due to a finite model capacity). Given this intuition, we find that it is often helpful to include more relevant and compatible tasks based on prior knowledge (*e.g.* VQA benefits from better object detection) or a systematic taskonomy evaluation.[3]

C. Additional ablation results

In Table C.1, we show more ablation results of our UniT model on the three datasets, COCO detection, SNLI-VE, and MNLI, under the same settings as in our ablation analyses in Sec. 4.2 and Table 4 in our main paper:

- **Image encoder hidden size:** Increasing the hidden size of the image encoder from 256 (default in DETR) to 768 (the BERT hidden size) leads to noticeably lower detection performance (line 2), which is possibly due to overfitting in the detection features.
- **Initializing convnet backbone from ImageNet**: Instead of initializing the convolutional network backbone in the image encoder from a detection-pretrained ResNet-50 in DETR [2], in this setting (line 3) the backbone is initialized from a ResNet-50 pretrained on ImageNet classification. It can be seen that the classification-pretrained backbone leads to lower COCO detection mAP. We suspect this is due to a relatively small number of training iterations on the COCO detection dataset – here we are using a total of 500k iterations on three datasets, while DETR [2] is trained for over 900k iterations (500 epochs) on the COCO dataset alone.
- **The number of queries in decoder:** In this setting, we vary the number of the query vectors in the decoder (*i.e.* the length of the query embedding sequence $\mathbf{q}^{task}$ in Sec. 3.3) on SNLI-VE and MNLI (while keeping a fixed number of 100 queries on the COCO detection task). We found that using only 1 query in the decoder (line 4) results in slightly lower accuracy on SNLI-VE, which is likely due to that the decoder needs to fuse multiple modalities in this case for visual entailment reasoning and benefits from more input queries. However, increasing the query number to 100 (line 5) does not give higher accuracy on SNLI-VE than the default setting (25 queries).
- **Learning rate:** We found that the joint training performance is sensitive to the learning rate. In line 6, training diverges with a higher learning rate (1e-4) than the default value of 5e-5. On the other hand, with a lower learning rate (1e-5) in line 7, the COCO detection mAP is noticeably lower while the SNLI-VE and MNLI accuracies are higher. These results show that different tasks have different optimal learning rates, which adds to the

[3] such as http://taskonomy.stanford.edu/

#	Model configuration	COCO det. mAP	SNLI-VE accuracy	MNLI-mm accuracy
1	UniT (default, d_t^d=768, N_d=6)	38.79	69.27	81.41
2	image encoder hidden size, d_v^e=768	33.39	68.53	81.01
3	initializing backbone from ImageNet instead of DETR	36.65	69.07	80.64
4	number of queries=1 for SNLI-VE and MNLI-mm	38.75	68.66	81.66
5	number of queries=100 for SNLI-VE and MNLI-mm	38.63	69.14	81.09
6	learning rate=1e-4	(training diverged in this setting)		
7	learning rate=1e-5	29.88	70.39	83.74
8	train for 1M iterations	39.96	69.31	79.88
9	init from COCO single-task	40.98	68.72	81.08
10	init from COCO single-task w/ frozen encoders	38.88	65.77	61.47
11	similar to 10 but do not init. detection class and box heads	37.18	65.01	59.87
12	similar to 10 but only freeze vision encoder	37.87	68.70	81.11

Table C.1: Additional ablation analyses of our UniT model with different model configurations on COCO detection, SNLI-VE, and MNLI (under the same settings as in Sec. 4.2 in the main paper).

difficulties of joint training. Our default setting (line 1) uses a 5e-5 learning rate as a balance across tasks. A possible future direction is to explore custom and adaptive learning rates on different components of the model.

- **More training iterations:** Using $2\times$ more training iterations (1M) yields higher COCO detection mAP but lower MNLI accuracy (line 8). We suspect it is because the detection task requires a longer training schedule to output a list of boxes and classes, while the MNLI dataset only requires a single classification prediction and too many iterations could cause overfitting.
- **Initialization from the COCO single-task model:** To provide more training iterations on the detection task, in line 9 we also experiment with initializing the multi-task model from the single-task model trained on the COCO detection dataset alone (*i.e*. COCO init. as described in Sec. 4.1 in the main paper). As expected, initializing from a COCO-pretrained single-task model leads to a noticeably higher detection mAP (line 9 vs 1), but we also see a slight performance drop on the other two datasets.
- **Freezing the encoders in UniT:** In multi-task training with UniT, the image and text encoders are jointly trained with the rest of the model. However, one might wonder whether it is necessary or beneficial to train these modality-specific encoders jointly. Is it possible to learn the encoders once on individual uni-modal tasks and directly use them on other tasks without retraining?

  In this setting, we experiment with pretrained and frozen encoders. In line 10, we initialize the image encoder from a single-task model pretrained on COCO detection (same as in line 9), initialize the text encoder from a pretrained BERT model (bert-base-uncased), and freeze both decoders during training. We also train another variant (line 11), which is similar to line 10 except that the detection class and box heads are randomly initialized.

  It can be seen that these two variants have significantly lower performance on all three datasets. In line 12, we still freeze the image encoder but update the text encoder (BERT) during training. It leads to better accuracy on MNLI and SNLI-VE that involve language understanding, but still relatively low detection mAP on COCO. These results suggest that it is hard to build a single shared decoder upon the frozen representations of each modality and that the co-adaptation of the decoder *and* the encoders is critical to multi-task training.

D. Learning curves

In Figure D.1, we show the learning curves of our unified model on all the 8 datasets with shared or separate decoders (Table 3 line 5 and 4 in the main paper), plotting the per-task performance on the validation data against training iterations. We also show the learning curves of the models trained on a single dataset (Table 3 line 1) for reference.

It can be seen that in our multi-task models, the performance of most tasks increases monotonically during training. However, SST-2 accuracy and QNLI accuracy reach their peak in early iterations and slightly decline as the training goes on, likely due to overfitting on these two relatively small datasets.

E. More visualizations

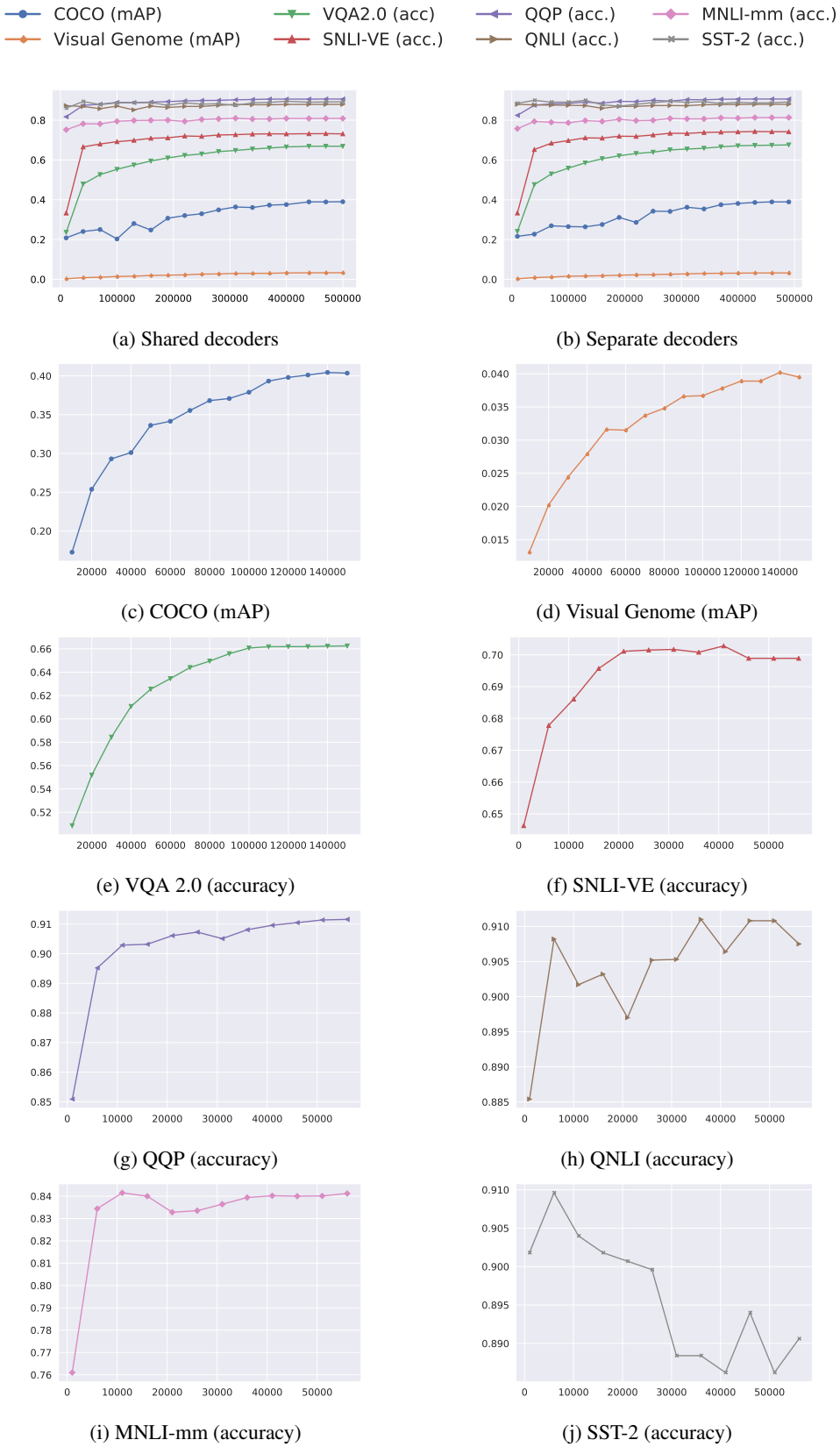
Figure E.1 shows additional predicted examples from our UniT model across 8 datasets (Table 3 line 5 in the main paper). The same model is applied to each task and dataset.

References

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of CVPR*, pages 6077–6086, 2018. 1

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of ECCV*, 2020. 2

[3] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 1



(a) Shared decoders

(b) Separate decoders

(c) COCO (mAP)

(d) Visual Genome (mAP)

(e) VQA 2.0 (accuracy)

(f) SNLI-VE (accuracy)

(g) QQP (accuracy)

(h) QNLI (accuracy)

(i) MNLI-mm (accuracy)

(j) SST-2 (accuracy)

Figure D.1: Learning curves of various experiments. The plots show the validation metrics at various iterations during the training process of (a) shared decoders, (b) separate decoders, and (c - j) single task training for each of the tasks.

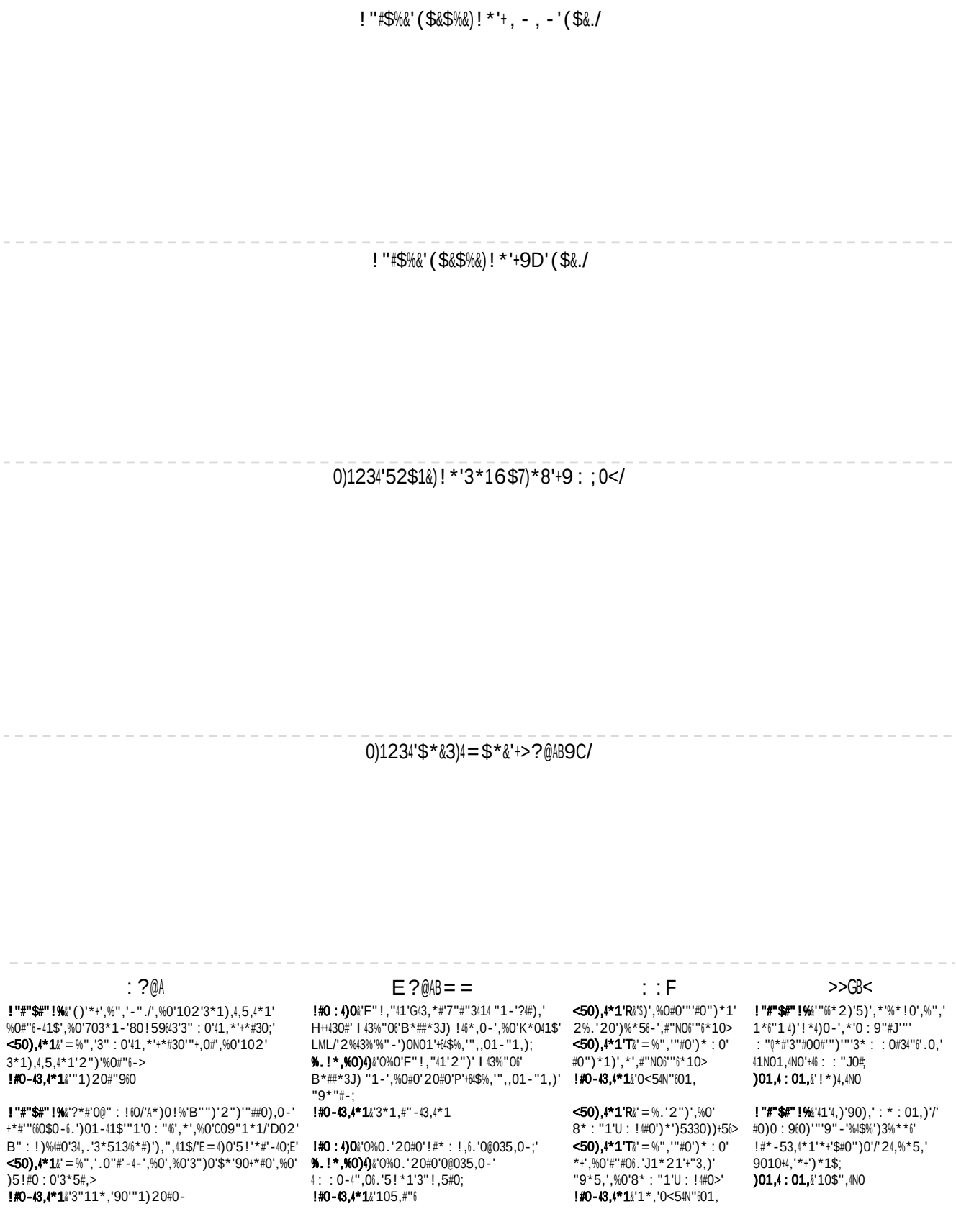


Figure E.1: More predictions of our model with a shared decoder (Table 3 line 5 in the main paper) across 8 datasets.