Transformer is All You Need: Multimodal Multitask Learning with a Unified Transformer

Ronghang Hu Amanpreet Singh
Facebook AI Research (FAIR)

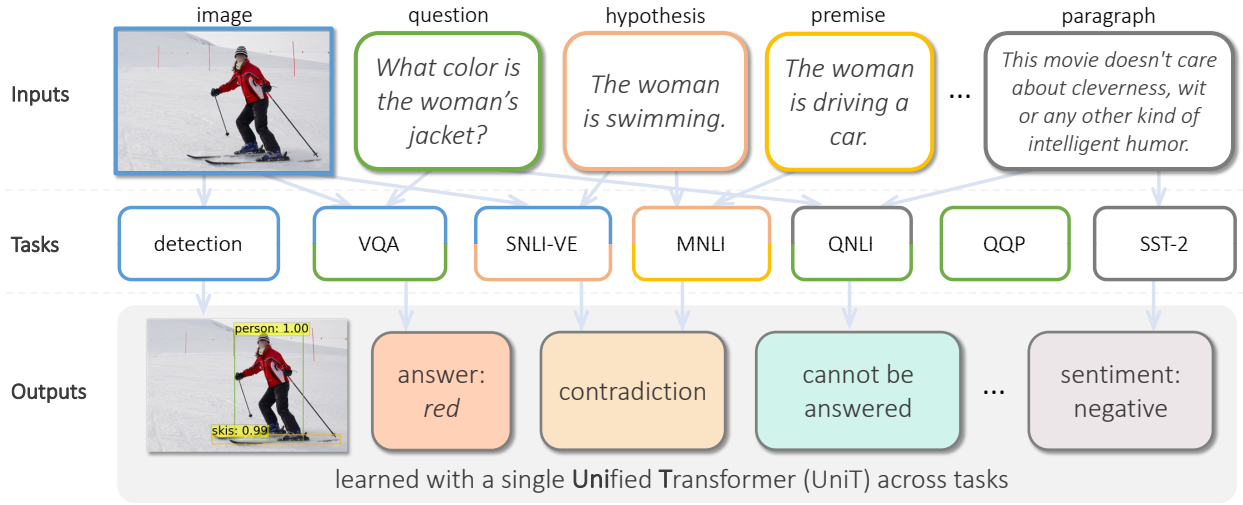


Figure 1: In this work, we propose UniT, which jointly learns multiple tasks across different modalities with a **Uni**fied **T**ransformer. Our UniT model simultaneously handles 7 tasks ranging from object detection to vision-and-language reasoning and language understanding, achieving strong performance on each task with a unified set of model parameters.

Abstract

We propose UniT, a Unified Transformer model to simultaneously learn the most prominent tasks across different domains, ranging from object detection to language understanding and multimodal reasoning. Based on the transformer encoder-decoder architecture, our UniT model encodes each input modality with an encoder and makes predictions on each task with a shared decoder over the encoded input representations, followed by task-specific output heads. The entire model is jointly trained end-to-end with losses from each task. Compared to previous efforts on multi-task learning with transformers, we share the same model parameters to all tasks instead of separately fine-tuning task-specific models and handle a much higher variety of tasks across different domains. In our experiments, we learn 7 tasks jointly over 8 datasets, achieving comparable performance to well-established prior work on each domain under the same supervision with a compact set of model parameters. Code will be released in MMF at https://mmf.sh.

1. Introduction

First proposed in [58], transformers have shown great success in a wide range of domains including but not limited to natural language, images, video, and audio. Previous works (*e.g.* [13, 42, 43, 4, 64, 34, 28, 44, 48]) demonstrate that transformers trained on large corpora learn strong representations for a wide range of downstream language tasks. In the visual domain, models based on transformers have achieved promising results on image classification, object detection, and panoptic segmentation (*e.g.* [39, 3, 21, 20, 46, 14, 60, 5, 71, 2, 57]). Besides modeling a single modality, transformer models also exhibit strong performance in joint vision-and-language reasoning tasks such as visual question answering (*e.g.* [30, 37, 38, 56, 9, 29, 55, 70, 22]).

However, despite the above achievements in application of transformers to *specific domains*, there has not been much prior effort to connect different tasks *across domains* with transformers. After witnessing the success of transformers, various questions naturally arise; could a transformer model trained for natural language inference on tex-

tual input also perform object detection on images, or could an image classifier based on transformers also check textual entailment? Overall, is it possible to build a single, unified model that *simultaneously* handles tasks in a variety of domains? Prior work tries to tackle some of these questions but only in limited scope:

- work only on tasks from a single domain or specific multimodal domains; ViT [14] and DETR [5] focus on vision-only tasks, BERT [13] and its derivative works [34, 64, 28, 44] only handle language tasks, while VisualBERT, VILBERT [37, 30] and other multimodal transformers work only on specific multimodal domain of vision and language.
- involves task-specific fine-tuning for each of the tasks, not leveraging any shared parameters across the tasks, usually ending up with N times the parameters for N tasks; one has to separately fine-tune a model for each of the tasks with BERT.
- performs multi-tasking upon related or similar tasks only from a single domain, sometimes with hard-coded training strategies; for example, T5 [44] works only on tasks in the language domain, while VILBERT-MT [38] works only on related vision-and-language tasks.

In this work, we build a **Uni**fied **T**ransformer **(UniT)** encoder-decoder model that takes images and/or text as inputs and jointly train on multiple tasks ranging from visual perception and language understanding to joint vision-*and*-language reasoning. UniT consists of encoding modules which encode each input modality as a sequence of hidden states (feature vectors), and a transformer decoder over the encoded input modalities, followed by task-specific output heads applied on the decoder hidden states to make the final predictions for each of the tasks. Compared to previous work on multi-task learning with transformers (*e.g.* [38]), we train UniT and achieve comparable performance to well-established prior work on a much larger variety of tasks; not only joint vision-and-language tasks such as VQA, but also vision-only as well as language-only tasks. We make the following contributions in this work:

- We propose **UniT**, a **uni**fied **t**ransformer encoder-decoder architecture capable of connecting and learning multiple tasks and domains in a single model.
- We jointly learn the most prominent tasks in the visual and textual domains and their intersections, namely object detection, visual question answering, visual entailment, and natural language understanding tasks in the GLUE benckmark [59], including QNLI [45], MNLI [61], QQP [23], and SST-2 [51]. We show that these diverse tasks can be learned simultaneously and converge properly under our training scheme.
- Through analyses across a variety of tasks, we show that multimodal tasks such as VQA and visual entailment benefit from multi-task training with uni-modal tasks.

2. Related work

Transformers on language, vision, and multimodal tasks. Transformers were first applied to the language domain for sequence-to-sequence modeling [58]. BERT [13], GPT [42, 43, 4], XLNet [64], RoBERTa [34], ALBERT [28], T5 [44], T-NLG [48] and other recent works show that transformers pretrained on large corpora learn language representations that can be transferred to a number of downstream tasks through fine-tuning.

In the visual domain, Image Transformer [39] builds a transformer model over local pixel neighborhoods for image generation and super-resolution. Image GPT [8] and ViT [14] apply transformers to flattened image pixels or image patches for classification. DETR [5] performs detection and segmentation with an end-to-end encoder-decoder model. In addition, the multi-head self-attention mechanism from transformers also benefits a wide range of vision applications (*e.g.* [60, 46, 11, 68, 69]). For joint vision-and-language reasoning tasks such as visual question answering, transformer models have been extended to take both the image and the text modalities as inputs (*e.g.* VisualBERT [30], VILBERT [37, 38], LXMERT [56], and UNITER [9]).

Most of these previous applications and extensions of transformers train (or fine-tune) a specific model for each of the tasks of interest. In BERT [13], a pretrained transformer model is fine-tuned separately on multiple downstream language tasks. In T5 [44], a text-to-text transformer is jointly pretrained on different language tasks. However, despite learning generic representations through multi-task pretraining, T5 still fine-tunes a different set of parameters for each downstream task. On the contrary, we simultaneously learn multiple tasks within a single transformer.

Multi-task learning with transformers. There has been a long history of work on multi-task learning [6, 12] in vision (*e.g.* [17, 67, 54, 53, 66]), language (*e.g.* [52, 16, 32, 49, 10]), or multimodal areas (*e.g.* [24, 25, 41, 7, 38]). Most previous efforts on multi-task learning focus on specific domains or modalities, often with model architectures tailored to the domain. However, there are also notable prior work on multi-task learning across domains with a single generic model. In [24], it is shown that an encoder-decoder architecture based on transformer's multi-head attention mechanism can be applied to different input and output domains such as image classification, machine translation, and image captioning. The decoders in [24] are specifically designed for each output task, while our model involves fewer task-specific details as we apply the same decoder architecture on all tasks. In MT-DNN [33], a multi-task language understanding model is built by sharing lower layers in a transformer while making the top layer task-specific. In VILBERT-MT [38], 12 vision-and-language tasks were jointly learned with a multi-task transformer model based

on VILBERT [37]. Compared to [33] and [38], we expand beyond fixed input modalities and jointly handle different single-modal (vision-only and language-only) and multi-modal tasks with a unified transformer model.

3. UniT: One transformer to learn them all

In this paper, we jointly learn multiple tasks across different modalities with a unified single model. Our model, UniT, is built upon the transformer encoder-decoder architecture [58, 5], consisting of separate encoders for each input modality type followed by a decoder (per-task or shared) with simple task-specific heads. Figure 2 shows an overview of UniT.

We consider two input modalities: images and text. For our transformer-based encoder on image inputs, inspired by [5], we first apply a convolutional neural network backbone to extract a visual feature map, which is further encoded by a transformer encoder into a list of hidden states to incorporate global contextual information. For language inputs, we use BERT [13], specifically the 12-layer uncased version, to encode the input words (*e.g.* questions) into a sequence of hidden states from BERT's last layer. After encoding input modalities into hidden state sequences, we apply the transformer decoder on either a single encoded modality or the concatenated sequence of both encoded modalities, depending on whether the task is uni-modal (*i.e.* vision-only or language-only) or multimodal. We explore either having separate (*i.e.* task-specific) or shared decoders among all tasks. Finally, the representation from the transformer decoder is passed to a task-specific head such as a simple two-layer classifier, which outputs the final predictions. Given the simplicity of UniT, it can be extended easily to more modalities and inputs.

We empirically show that our model can jointly learn 7 different tasks on 8 datasets. The following sections further describe the details of each component in UniT.

3.1. Image encoder

The vision-only tasks (such as object detection) and vision-and-language tasks (such as visual question answering and visual entailment) require perceiving and understanding an image I as input. In our model, we encode the input image I with a convolutional neural network followed by a transformer encoder, into a list of encoded visual hidden states $\mathbf{h}^v = \{h_1^v, h_2^v, \cdots, h_L^v\}$.

Our image encoding process is inspired by and similar to DETR [5]. First, a convolutional neural network backbone B is applied on the input image to extract a visual feature map $\mathbf{x}^v$ of size $H_v \times W_v \times d_v^b$ as

$$\mathbf{x}^v = B(I). \tag{1}$$

In our implementation, the backbone network B follows the structure of ResNet-50 [18] with dilation [65] applied to its last C5 block, and is pretrained on object detection in [5].

We apply a visual transformer encoder E_v with N_v layers and hidden size d_v^e on top of the feature map $\mathbf{x}^v$ to further encode it to visual hidden states $\mathbf{h}^v$ of size $L \times d_v^e$ (where $L = H_v \times W_v$ is the length of the encoded visual hidden states). In addition, given that different tasks (such as object detection and VQA) might require extracting different types of information, we also add a task embedding vector w_v^{task} into the transformer encoder to allow it to extract task-specific information in its output as follows.

$$\mathbf{h}^v = \{h_1^v, h_2^v, \cdots, h_L^v\} = E_v(P_{b\to e}(\mathbf{x}^v), w_v^{task}) \tag{2}$$

$P_{b\to e}$ is a linear projection from visual feature dimension d_v^b to encoder hidden size d_v^e. The structure of the visual transformer encoder E_v follows DETR [5], where positional encoding is added to the feature map. The task token w^{task} is a learned parameter of dimension d_v^e, which is concatenated to the beginning of the flattened visual feature list $P_{b\to e}(\mathbf{x}^v)$ and stripped from the output hidden states $\mathbf{h}^v$.

3.2. Text encoder

GLUE benchmark [59] tasks such as QNLI [45], MNLI [61], QQP [23], and SST-2 [51] as well as the joint vision-and-language reasoning tasks such as VQA and visual entailment provide a textual input. We encode the textual input using BERT [13] – a transformer encoder model pretrained on large corpora with masked language modeling and next sentence prediction tasks.

Given the input text (*e.g.* a sentence or a pair of sentences), we tokenize it in the same way as in BERT into a sequence of S tokens $\{w_1, \cdots, w_S\}$, with $w_1 =$ [CLS] (the special pooling token in BERT for classification). The token sequence is then used as input to a pretrained BERT model to extract a sequence of textual hidden states $\mathbf{h}^t$ of size $S \times d_t^e$, where d_t^e is the BERT hidden size. Similar to the image encoder, in the text encoder, we also add a learned task embedding vector w_t^{task} as part of the BERT input by prefixing it at the beginning of the embedded token sequence, and later stripping it from the output text hidden states as follows.

$$\mathbf{h}^t = \{h_1^t, h_2^t, \cdots, h_S^t\} = \text{BERT}(\{w_1, \cdots, w_S\}, w_t^{task}) \tag{3}$$

However, we find that it works nearly equally well in practice to keep only the hidden vector corresponding to [CLS] in $\mathbf{h}^t$ as input to the decoder, which saves computation.

In our implementation, we use a pretrained BERT-base uncased model from the Huggingface's Transformers library [62], which has $d_t^e = 768$ and $N_t = 12$ layers.

3.3. Domain-agnostic UniT decoder

After encoding the input modalities, we apply on them a transformer decoder D with hidden size d_t^d and number

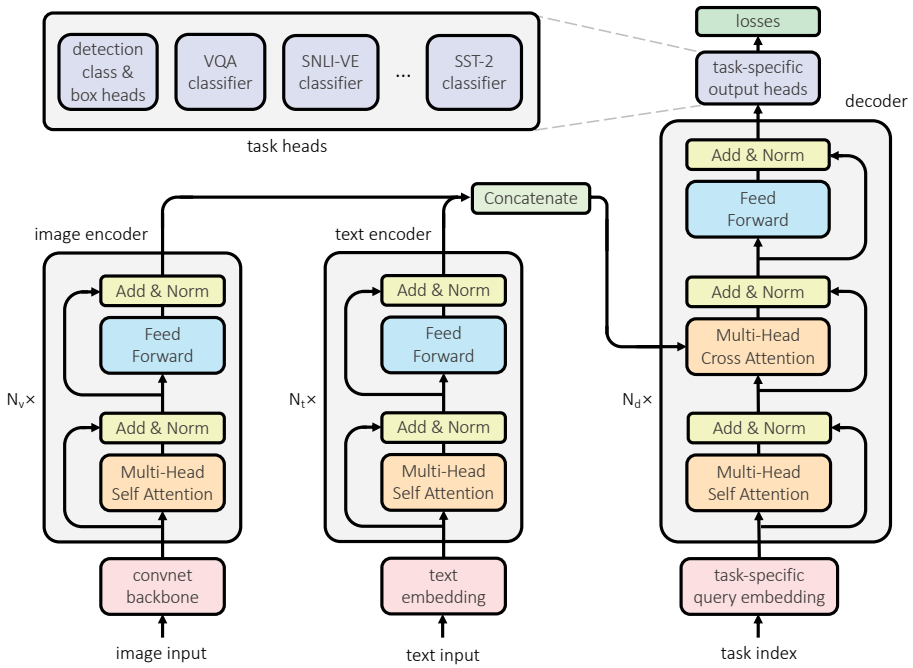


Figure 2: An overview of our UniT model, which jointly handles a wide range of tasks in different domains with a unified transformer encoder-decoder architecture. Our model uses an image encoder to encode the visual inputs (Sec. 3.1), a text encoder to encode the language inputs (Sec. 3.2), and a joint decoder with per-task query embedding (Sec. 3.3) followed by task-specific heads (Sec. 3.4) to make the final outputs for each task.

of layers N_d to output a sequence of decoded hidden states $\mathbf{h}^{dec}$ for predictions on each task. Unlike the image and text encoders with specific architectural designs for each modality, our decoder is built upon the same domain-agnostic transformer decoder architecture [58] across all tasks.

For vision-only tasks, we apply the decoder on the encoded image $\mathbf{h}^{enc} = \mathbf{h}^{v}$ described in Sec. 3.1, for language-only tasks, we apply the decoder on the encoded text $\mathbf{h}^{enc} = \mathbf{h}^{t}$ in Sec. 3.2, and finally for joint vision-and-language tasks, we concatenate the encoded inputs from both modalities into a single sequence $\mathbf{h}^{enc} = \text{concat}(\mathbf{h}^{v}, \mathbf{h}^{t})$ as the input to the decoder.

The transformer decoder D takes the encoded input sequence $\mathbf{h}^{enc}$ and a task-specific query embedding sequence $\mathbf{q}^{task}$ of length q. It outputs a sequence of decoded hidden states $\mathbf{h}^{dec,l}$ for each of the l-th transformer decoder layer, which has the same length q as the query embedding $\mathbf{q}^{task}$.

$$\left\{\mathbf{h}^{dec,l}\right\} = D(\mathbf{h}^{enc}, \mathbf{q}^{task}) \tag{4}$$

In our implementation, we use the same transformer decoder architecture as in DETR [5]. In the l-th decoder layer, self-attention is applied among the decoder hidden states $\mathbf{h}^{dec,l}$ at different positions and cross-attention is applied to the encoded input modalities $\mathbf{h}^{enc}$.

In our experiments, with use either (i) a single shared decoder D^{all} for all tasks or (ii) a separate decoder D_i^{task} for each specific task i.

3.4. Task-specific output heads

On the decoder hidden states $\left\{\mathbf{h}^{dec,l}\right\}$, we apply a task-specific prediction head for each task t for final predictions.

For object detection, we add a class head to produce a classification output (including "background") and a box head to produce a bounding box output for each of the positions in $\{1, \ldots, q\}$ in the decoder hidden states. The class head and the box head follow the implementation in DETR [5]. For datasets with attribute labels on each box (the Visual Genome dataset [27] in our experiments), we also add an attribute classification head following the implementation of BUTD [1]. Each position in the decoder hidden states either produces an object class or background.

The outputs from the class and box heads are post-processed into object bounding boxes. Following DETR, we apply these heads to all layers l in the decoder hidden states $\mathbf{h}^{dec,l}$ during training as

$$\mathbf{c}^{l} = \text{class\_head}(\mathbf{h}^{dec,l}) \tag{5}$$

$$\mathbf{b}^{l} = \text{box\_head}(\mathbf{h}^{dec,l}) \tag{6}$$

$$\mathbf{a}^{l} = \text{attr\_head}(\mathbf{h}^{dec,l}, \mathbf{c}^{l}) \tag{7}$$

where $\mathbf{c}^{l}$, $\mathbf{b}^{l}$, and $\mathbf{a}^{l}$ are class, box and attribute output sequences, all having the same length q as the query embedding $\mathbf{q}^{task}$ for detection.

At test time, we only take the prediction from the top decoder layer, $\mathbf{h}^{dec,N_d}$. Since different detection datasets often have different numbers of classes, when training on multiple detection datasets, each dataset has its own class, box,

and attribute heads. We apply the same detection losses on the outputs $\mathbf{c}^l$ and $\mathbf{b}^l$ as in DETR, and the same attribute losses on $\mathbf{a}^l$ as in BUTD [1].

All other tasks that we address in this work, including visual question answering, visual entailment, and natural language understanding (QNLI, QQP, MNLI, and SST-2) can be cast as a classification task among c_t classes for task t. We apply a task-specific classifier on the first output position hidden state $\mathbf{h}_1^{dec,top}$ from the top decoder layer to output a classification prediction $\mathbf{p}$ of size c_t for the task t.

For the classifier, we use a two-layer perceptron with GeLU activation [19] (followed by dropout) and hidden dimension equal to decoder hidden size to generate the predictions. We apply the cross-entropy classification loss on the predictions $\mathbf{p}$ with ground-truth targets $\mathbf{t}$ to train the model.

$$\mathbf{p} = \mathrm{W}_1 \cdot \mathrm{GeLU}(\mathrm{W}_2 \cdot \mathbf{h}_1^{dec,top} + \mathrm{b}_2) + \mathrm{b}_1 \quad (8)$$

$$\mathrm{loss} = \mathrm{CrossEntropyLoss}(\mathbf{p}, \mathbf{t}) \quad (9)$$

3.5. Training

We jointly train UniT on multiple tasks. At each iteration during training, we randomly select a task and a dataset to fill a batch of samples. We manually specify a sampling probability for each task based on the dataset size and empirical evidence. In our implementation, we train with a batch size of 64 on 64 Nvidia Volta V100-SXM2-32GB GPUs (batch size 1 per GPU) in a distributed fashion, using the MMF framework [50] based on PyTorch [40].

We use the weighted Adam optimizer [26, 36] with a learning rate of 5e-5 and the warm-up cosine learning rate schedule [35] (using 2000 warm-up iterations). The optimizer updates the model parameters based on gradients from the task losses.[1]

We apply the scale and crop augmentation following DETR [5] on image inputs during training for object detection. On a detection training batch, an input image is randomly resized such that its shortest side is between 480 and 800 pixels, and then a crop with random width and height between 384 and 600 pixels is taken from the resized image. However, we do not apply scale and crop augmentation on vision-and-language tasks such as VQA, as these tasks often require the entire image for global reasoning (*e.g.* answering "how many people are there in the image" requires counting every person in the entire image). At test time for object detection and at both training and test time for vision-and-language tasks, an input image is resized to have a deterministic shortest side of 800 pixels.

[1]We update all parameters in the model in our default setting, even if some parameters are not used in the forward pass of a batch and their gradients remain zero (*e.g.* the VQA classifier parameters do not accumulate gradients on a detection batch but are still updated by the optimizer).

4. Experiments

To provide a thorough analysis of UniT and a fair comparison to established prior methods, we experiment with jointly learning prominent tasks from different domains, including object detection as a vision-only task, language understanding tasks from GLUE benchmark as language-only tasks, and visual reasoning tasks for joint vision-and-language understanding. For the object detection task, we use the COCO dataset [31] as a benchmark and also the Visual Genome (VG) dataset [27], which contains object classes as well as their attributes. For language understanding, we experiment with four tasks from the GLUE benchmark [59]: QNLI [45], QQP [23], MNLI-mismatched [61], and SST-2 [51]. For joint vision-and-language reasoning, we use the VQAv2 dataset [15] (with questions from Visual Genome [27] as additional training data) and also experiment with SNLI-VE [63], which requires classifying an image and sentence pair into whether the sentence entails, contradicts or is neutral with respect to the image. These datasets are used for pure research purpose only.

We experiment with two settings. First, we jointly train our model on object detection and VQA tasks in Sec. 4.1. Then, we further include language understanding tasks and an additional joint vision-and-language reasoning task (SNLI-VE) in Sec. 4.2.

4.1. Multitask learning on detection and VQA

We first experiment with training on the object detection task as a vision-only task and the visual question answering task that requires jointly modeling both the image and the text modalities.

Removing overlap. For object detection, we use the COCO detection dataset (COCO det.) [31] and the object annotations in the Visual Genome dataset (VG det.) [27]. For the VQA task, we use the VQAv2 dataset [15]. We split these datasets according to COCO train2017 and val2017 splits: for COCO detection, we use its train2017 split for training and val2017 split for evaluation; for other datasets (Visual Genome detection and VQAv2), we train on those images not overlapping with COCO val2017 and evaluate on those images in COCO val2017. We also use those

#	Experiment setup	COCO det. mAP	VG det. mAP	VQAv2 accuracy
1	single-task	40.4 / –	4.02	66.25 / –
2	separate	40.7 / –	4.22	**68.36** / –
3	shared	38.5 / –	4.16	61.51 / –
4	shared (COCO init.)	**40.9** / 41.2	**4.56**	67.72 / 68.43

Table 1: Performance of UniT on multi-task training over object detection and VQA. On the COCO det. and VQAv2 datasets, we also evaluate on the test-dev splits for our best model with shared decoders (line 4).

#	trained on			decoder	COCO det. mAP	VG det. mAP	VQAv2 accuracy
	COCO det.	VG det.	VQAv2				
1	single-task training			–	40.44	4.02	66.25
2	✓		✓	separate	39.30	–	67.16
3		✓	✓	separate	–	3.86	68.35
4	✓	✓	✓	separate	40.67	4.22	**68.36**
5	✓		✓	shared (COCO init.)	39.99	–	66.10
6		✓	✓	shared (COCO init.)	–	4.04	68.28
7	✓	✓	✓	shared (COCO init.)	**40.84**	**4.56**	67.72

Table 2: **Analyses on object detection and VQA with our UniT model**, using separate or shared decoders on different dataset combinations. The jointly trained model (line 7) outperforms the single-task models (line 1) on all the three datasets.

questions from the Visual Genome VQA dataset (on images not overlapping with COCO val2017) as additional training data, added to the training split of VQAv2.

Training. We train and evaluate our model under different combinations of tasks and datasets: COCO detection (COCO det.) + VQAv2, Visual Genome detection (VG det.) + VQAv2, and all the three datasets together. We also train it on a single dataset as a comparison. In each training combination, we experiment with two settings in our transformer decoder: 1) separate decoders on different tasks (without sharing decoder parameters) and 2) a single shared decoder for all tasks. Following previous work in these two areas, we evaluate the detection performance with mean average precision (mAP) and the VQA task with VQA accuracy.[2] During joint training, we sample all datasets with equal probability. We train for a total of 150k, 300k, and 450k iterations for experiments on one, two, and three datasets, respectively.[3]

Results. Table 1 shows the performance of our model jointly trained on the three datasets with separate (line 2) or shared decoders (line 3), and also the single-task performance of our model trained separately on each dataset (line 1). With separate decoders, our model trained jointly on the three datasets outperforms its counterparts with single-task training on all the three datasets. However, comparing line 3 with 1, we observe that while the joint model trained with shared decoders achieves non-trivial performance on the three datasets, it underperforms the single-task models on COCO detection and VQAv2 by a noticeable margin.

The object detection task requires structural outputs (bounding boxes with class labels, as opposed to a classification output in VQA), and the decoder needs to properly model the relations between different objects (such as their overlap to learn non-maximum suppression). Hence, object detection may require a longer training schedule, especially for shared decoders, to learn the complex behavior that models both the object relation in detection and the multimodal fusion and reasoning in VQA. To provide more training iterations on the detection task in the shared decoder setting, we experiment with initializing our model from a model trained on COCO detection alone (**COCO init.**) to continue training it on the joint tasks. In this case, the image encoder (including the convolutional network backbone and the transformer encoder in it) and the detection heads are initialized from the single-task COCO detection model in Table 1 line 1.

This variant of the joint model (in Table 1 line 4) with shared decoders outperforms single-task models on all the three datasets (line 1). Also, comparing with line 3, it can be seen that initialization from the COCO single-task model benefits on all the three datasets.

Ablations. We further evaluate with training on one dataset from each task (using either COCO for Visual Genome as the detection dataset). The results are shown in Table 2, where it can be seen that i) joint training on two detection datasets usually benefits both datasets (line 4 vs 2, line 4 vs 3, line 7 vs 5, and line 7 vs 6) and ii) training on VG + VQAv2 gives better VQA accuracy than training on COCO + VQAv2, which is likely due to the fact that the Visual Genome dataset contains a more diverse set of object annotations (attributes) and better coverage of visual concepts for visual question answering.

4.2. A Unified Transformer for multiple modalities

To further test the capabilities of UniT, we extend the training to 8 datasets, adding 4 language-only tasks from GLUE benchmark (QNLI, QQP, MNLI, and SST-2) and a vision-and-language dataset SNLI-VE for visual entailment. We show that UniT can jointly perform on all 7 tasks across 8 datasets competitively using 8× fewer parameters than task-specific fine-tuned similar models. Our final UniT model in Table 3 line 5 has around 201M parameters.

Training. For COCO, VQA and Visual Genome, we follow the splits created in the previous section and for SNLI-

[2] https://visualqa.org/evaluation.html

[3] When training on three datasets jointly with shared decoders, we empirically find that skipping optimizer updates (including momentum accumulation) on unused parameters with zero gradients (*e.g.* VQA classifier weights in a detection iteration) works better than updating all parameters. The latter often causes divergence, possibly because accumulating zero gradients leads to unstable momentum.

#	decoder	COCO det.	VG det.	VQAv2	SNLI-VE	QNLI	MNLI-mm	QQP	SST-2
1	UniT – single-task training	40.4	4.02	66.25 / –	70.52 / –	91.62 / –	84.23 / –	91.18 / –	91.63 / –
2	UniT – separate	32.2	2.54	67.38 / –	74.31 / –	87.68 / –	81.76 / –	90.44 / –	89.40 / –
3	UniT – shared	33.8	2.69	67.36 / –	74.14 / –	87.99 / –	81.40 / –	90.62 / –	89.40 / –
4	UniT – separate (COCO init.)	38.9	3.22	67.58 / –	74.20 / –	87.99 / –	81.33 / –	90.61 / –	89.17 / –
5	UniT – shared (COCO init.)	39.0	3.29	66.97 / 67.03	73.16 / 73.16	87.95 / 88.0	80.91 / 79.8	90.64 / 88.4	89.29 / 91.5
6	DETR [5]	43.3	4.02	–	–	–	–	–	–
7	VisualBERT [30]	–	–	67.36 / 67.37	75.69 / 75.09	–	–	–	–
8	BERT [13] (bert-base-uncased)	–	–	–	–	91.25 / 90.4	83.90 / 83.4	90.54 / 88.9	92.43 / 93.7

Table 3: **Performance of our UniT model on 7 tasks across 8 datasets**, ranging from vision-only tasks (object detection on COCO and VG), vision-and-language reasoning (visual question answering on VQAv2 and visual entailment on SNLI-VE), and language-only tasks from the GLUE benchmark (QNLI, MNLI, QQP, and SST-2). For the line 5, 7 and 8, we also show results on VQAv2 test-dev, SNLI-VE test, and from GLUE evaluation server.

VE and the GLUE tasks we follow the official splits.[4][5] Similar to Sec. 4.1, we experiment with three different settings: (i) single-task training where each model is trained separately on each task, (ii) multi-task training with separate decoders where the model has a specific decoder for each task but is jointly trained on all of the tasks, and (iii) multi-task training same as (ii) but with a shared decoder instead of separate ones. In (iii), the model still contains task-specific heads for each task to generate predictions as explained in Sec. 3.4. Following Sec. 4.1, we also train a variation of (ii) and (iii), where we initialize the image encoder and the decoder from a single task COCO-pretrained UniT model. We train all models for 500k iterations and keep the rest of the hyper-parameters the same as in previous experiments.

Results. Table 3 shows the performance of UniT under different variants as discussed above. UniT models trained on each task separately (line 1) outperform all other variants on all tasks except multimodal tasks VQAv2 and SNLI-VE. This is unsurprising as (i) the unimodal tasks have low cross-modality overlap, (ii) each task is trained for full 500k iteration, compared to some percentage of 500k in joint training, and (iii) for 8 tasks, there are a total of 8× parameters compared to the shared decoder (line 3 and 5). On the other hand, we see that vision-and-language tasks, namely VQAv2 and SNLI-VE, consistently benefit from multi-task training together with vision-only and language-only tasks across different settings, suggesting that learning better unimodal representations also benefits multimodal reasoning.

We compare our approach to well-established domain-specific methods based on transformer on each task. For object detection on COCO and VG, we compare to DETR [5] (line 6), a recent transformer-based approach for object detection from which our image encoder is inspired. For joint vision-and-language reasoning (visual question answering on VQAv2 and visual entailment on SNLI-VE), we compare to VisualBERT [30] (line 7), which extends the BERT model [13] to also take detected objects as inputs.[6] Note that VisualBERT relies on an external Faster R-CNN object detector [47] to extract objects as visual representations, whereas our model directly uses the raw image pixels as input. On natural language understanding tasks from the GLUE benchmark, we compare to BERT [13] (line 8).[7] From Table 4, it can be seen that our model achieves strong performance on each task with a single generic model. Despite that there is still a gap when comparing line 5 to line 6, 7 and 8, our model shows promising results approaching these domain-specific transformer-based models, especially considering that DETR, VisualBERT and BERT have hyperparameters (such as learning rate and training schedule) tailored to each domain, while our model adopts the same hyperparameters across all 8 datasets. Figure 3 shows the predictions of our model (in Table 4 line 5) on each dataset.

Ablations. To better understand the effect of each hyperparameter on multi-modal multi-task training with UniT,

[4]GLUE tasks were downloaded from https://gluebenchmark.com/tasks
[5]SNLI-VE was acquired from https://github.com/necla-ml/SNLI-VE

#	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	decoder layer number, N_d=8	40.13	68.17	80.58
3	decoder layer number, N_d=12	39.02	68.82	81.15
4	decoder hidden size, d_t^d=256	36.32	69.68	81.09
5	using all hidden states from BERT instead of just [CLS]	38.24	69.76	81.31
6	losses on all decoder layers for SNLI-VE and MNLI-mm	39.46	69.06	81.67
7	no task embedding tokens	38.61	70.22	81.45
8	batch size = 32	35.03	68.57	79.62

Table 4: Ablation analyses of our UniT model with different model configurations on COCO det., SNLI-VE, and MNLI.

[6]We compare to the variant of VisualBERT without masked language modeling pretraining on vision-and-language datasets for fair comparison.
[7]For fair comparison, we fine-tune and reevaluate `bert-base-uncased` checkpoint from the Transformers library [62] on GLUE evaluation server.

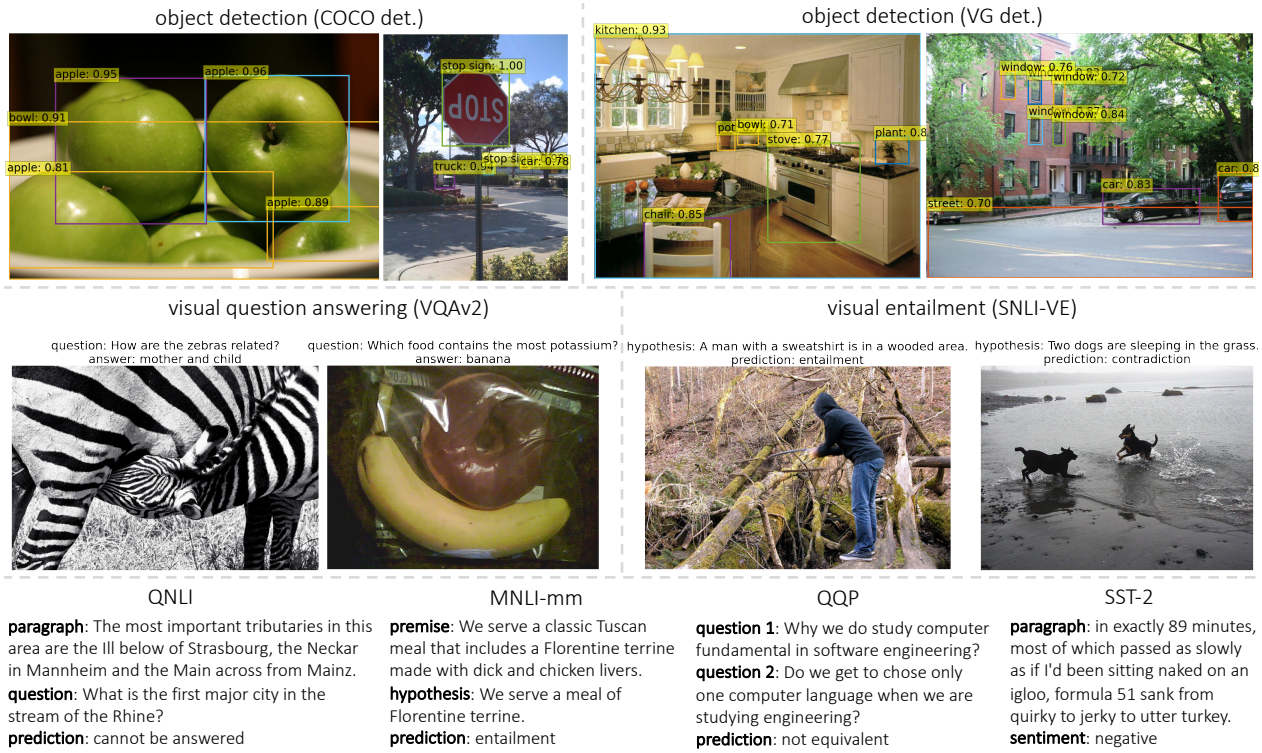


Figure 3: Predictions of our model with a shared decoder (Table 3 line 5) across 8 datasets. Our model jointly handles a large variety of tasks above through a unified transformer encoder-decoder architecture.

we conduct extensive ablations shown in Table 4. We choose a subset of tasks which have the potential of improving by training jointly: COCO object detection, SNLI-VE, and MNLI. We choose these three tasks as MNLI-mismatched and SNLI-VE are related tasks involving natural language inference at the core, and SNLI-VE and COCO share the image source *i.e.* Flickr. The results are as follows (see supplemental for more ablation analyses).

- **Decoder layers and hidden size:** There is a drop in detection mAP with a smaller decoder hidden size (line 4), while it does not hurt SNLI-VE or MNLI-mm. This is likely because COCO is a larger dataset with 1.5 million object instances and benefits from larger models. The analyses on decoder layer number N_d (line 2 and 3) confirms this intuition as $N_d = 8$ gives better detection mAP. Meanwhile, doubling the decoder layers to $N_d = 12$ does not help detection as much, probably due to overfitting with very large models. In addition, we find that too large decoder hidden size ($d_t^d = 1536$) could lead to divergence in detection training.
- **All hidden states from BERT**: Using all BERT outputs as input to the decoder (instead of just the [CLS] token as in Sec. 3.2) has a relatively minor (and mixed) impact on the performance while increasing computation cost (line 5), suggesting that the pooled vector from BERT should be sufficient for most downstream tasks.
- **Losses on all decoder layers**: While losses on intermediate layer outputs benefit object detection (as shown in [5]), it does not benefit SNLI-VE or MNLI (line 6), likely because these tasks only require outputting a single label, unlike dense detection outputs.
- **No task embedding tokens**: We find that removing the task embedding from the encoders (line 7) does not hurt performance, probably because the image encoder can extract generic (instead of task-specific) visual representations applicable to both COCO and SNLI-VE, and likewise for the language encoder.
- **Batch size and learning rate:** A smaller batch size (line 8) leads to lower performance. Also, using a larger learning rate (1e-4 as in DETR [5] and MLM in BERT [13]) often causes divergence in joint training, while our smaller 5e-5 learning rate provides stable training.

5. Conclusion

In this work, we show that the transformer framework can be applied over a variety of domains to jointly handle multiple tasks within a single unified encoder-decoder model. Our UniT model simultaneously addresses 7 tasks across 8 datasets and achieves strong performance on each task with a single set of shared parameters. Through a domain-agnostic transformer architecture, our model makes a step towards building general-purpose intelligence agents capable of handling a wide range of applications in different domains, including visual perception, language understanding, and reasoning over multiple modalities.

Acknowledgements

We are grateful to Devi Parikh, Douwe Kiela, Marcus Rohrbach, Vedanuj Goswami, and other colleagues at FAIR for fruitful discussions and feedback.

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. 4, 5

[2] Josh Beal, Eric Kim, Eric Tzeng, Dong Huk Park, Andrew Zhai, and Dmitry Kislyuk. Toward transformer-based object detection. *arXiv preprint arXiv:2012.09958*, 2020. 1

[3] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3286–3295, 2019. 1

[4] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1, 2

[5] 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. 1, 2, 3, 4, 5, 7, 8, 12

[6] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997. 2

[7] Devendra Singh Chaplot, Lisa Lee, Ruslan Salakhutdinov, Devi Parikh, and Dhruv Batra. Embodied multimodal multitask learning. *arXiv preprint arXiv:1902.01385*, 2019. 2

[8] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020. 2

[9] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision*, pages 104–120. Springer, 2020. 1, 2

[10] Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D Manning, and Quoc V Le. Bam! born-again multi-task networks for natural language understanding. *arXiv preprint arXiv:1907.04829*, 2019. 2

[11] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. *arXiv preprint arXiv:1911.03584*, 2019. 2

[12] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020. 2

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, 2019. 1, 2, 3, 7, 8

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

[15] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of CVPR*, pages 6904–6913, 2017. 5

[16] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*, 2016. 2

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of CVPR*, pages 770–778, 2016. 3

[19] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *arXiv preprint arXiv:1606.08415*, 2016. 5

[20] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3464–3473, 2019. 1

[21] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. *Advances in neural information processing systems*, 31:9401–9411, 2018. 1

[22] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9992–10002, 2020. 1

[23] Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. First Quora dataset release: Question pairs. https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs. Jan 2017. 2, 3, 5

[24] Lukasz Kaiser, Aidan N Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. One model to learn them all. *arXiv preprint arXiv:1706.05137*, 2017. 2

[25] Douwe Kiela, Alexis Conneau, Allan Jabri, and Maximilian Nickel. Learning visually grounded sentence representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 408–418, 2018. 2

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[27] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense

image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 4, 5

[28] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019. 1, 2

[29] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, pages 11336–11344, 2020. 1

[30] 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, 2, 7

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of ECCV*, pages 740–755. Springer, 2014. 5

[32] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Adversarial multi-task learning for text classification. *arXiv preprint arXiv:1704.05742*, 2017. 2

[33] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In *Proceedings of ACL*, 2019. 2, 3

[34] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 1, 2

[35] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 5

[37] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proceedings of NeurIPS*, pages 13–23, 2019. 1, 2, 3

[38] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of CVPR*, pages 10437–10446, 2020. 1, 2, 3

[39] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. *arXiv preprint arXiv:1802.05751*, 2018. 1, 2

[40] 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, *Proceedings of NeurIPS*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[41] Subhojeet Pramanik, Priyanka Agrawal, and Aman Hussain. Omninet: A unified architecture for multi-modal multi-task learning. *arXiv preprint arXiv:1907.07804*, 2019. 2

[42] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. *Technical report, OpenAI*, 2018. 1, 2

[43] Alec Radford, Jeffrey Wu, Dario Amodei, Daniela Amodei, Jack Clark, Miles Brundage, and Ilya Sutskever. Better language models and their implications. *OpenAI Blog https://openai. com/blog/better-language-models*, 2019. 1, 2

[44] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 1, 2

[45] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP/IJCNLP*, pages 2383–2392. Association for Computational Linguistics, 2016. 2, 3, 5

[46] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Standalone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019. 1, 2

[47] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 7

[48] C Rosset. Turing-NLG: A 17-billion-parameter language model by microsoft. *Microsoft Blog*, 2020. 1, 2

[49] Victor Sanh, Thomas Wolf, and Sebastian Ruder. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6949–6956, 2019. 2

[50] Amanpreet Singh, Vedanuj Goswami, Vivek Natarajan, Yu Jiang, Xinlei Chen, Meet Shah, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Mmf: A multimodal framework for vision and language research. https://github.com/facebookresearch/mmf, 2020. 5

[51] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP/IJCNLP*, pages 1631–1642, 2013. 2, 3, 5

[52] Anders Søgaard and Yoav Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, 2016. 2

[53] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, pages 9120–9132. PMLR, 2020. 2

[54] Gjorgji Strezoski, Nanne van Noord, and Marcel Worring. Many task learning with task routing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1375–1384, 2019. 2

[55] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 1

[56] Hao Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of EMNLP/IJCNLP*, 2019. 1, 2

[57] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 1

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of NeurIPS*, pages 5998–6008, 2017. 1, 2, 3, 4

[59] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of ICLR*, 2019. 2, 3, 5

[60] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of CVPR*, pages 7794–7803, 2018. 1, 2

[61] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL-HLT*, 2018. 2, 3, 5

[62] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020. 3, 7

[63] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*, 2019. 5

[64] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Proceedings of NeurIPS*, pages 5753–5763, 2019. 1, 2

[65] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 3

[66] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust learning through cross-task consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11197–11206, 2020. 2

[67] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of CVPR*, pages 3712–3722, 2018. 2

[68] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10076–10085, 2020. 2

[69] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. *arXiv preprint arXiv:2012.09164*, 2020. 2

[70] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. In *AAAI*, pages 13041–13049, 2020. 1

[71] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1

Transformer is All You Need: 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. By default, we update all parameters in the model during training even if some parameters are not used in the forward pass of a batch and their gradients remain zero. However, we empirically find that this strategy sometimes causes 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 det., VG det., and VQAv2 with a shared decoder (Sec. 4.1 in the main paper), divergence happens with the default strategy (updating all parameters in 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. Additional ablation results

In Table D.1, we show more ablation results of our UniT model on the three datasets, COCO det., 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 [5], 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 [5] 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 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). 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

#	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)	0.30	–	–	0.50	–	0.20	–	–
3	ablation study (Sec. 4.2)	0.20	0.07	0.26	0.12	0.10	0.10	0.10	0.05

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

show that different tasks have different optimal learning rates, which adds to the difficulties of joint training. Our default setting (line 1) uses 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× 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 the 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.

#	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, q for SNLI-VE and MNLI-mm	38.75	68.66	81.66
5	number of queries=100, q 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 B.1: Additional ablation analyses of our UniT model with different model configurations on COCO det., SNLI-VE, and MNLI (under the same settings as in Sec. 4.2).

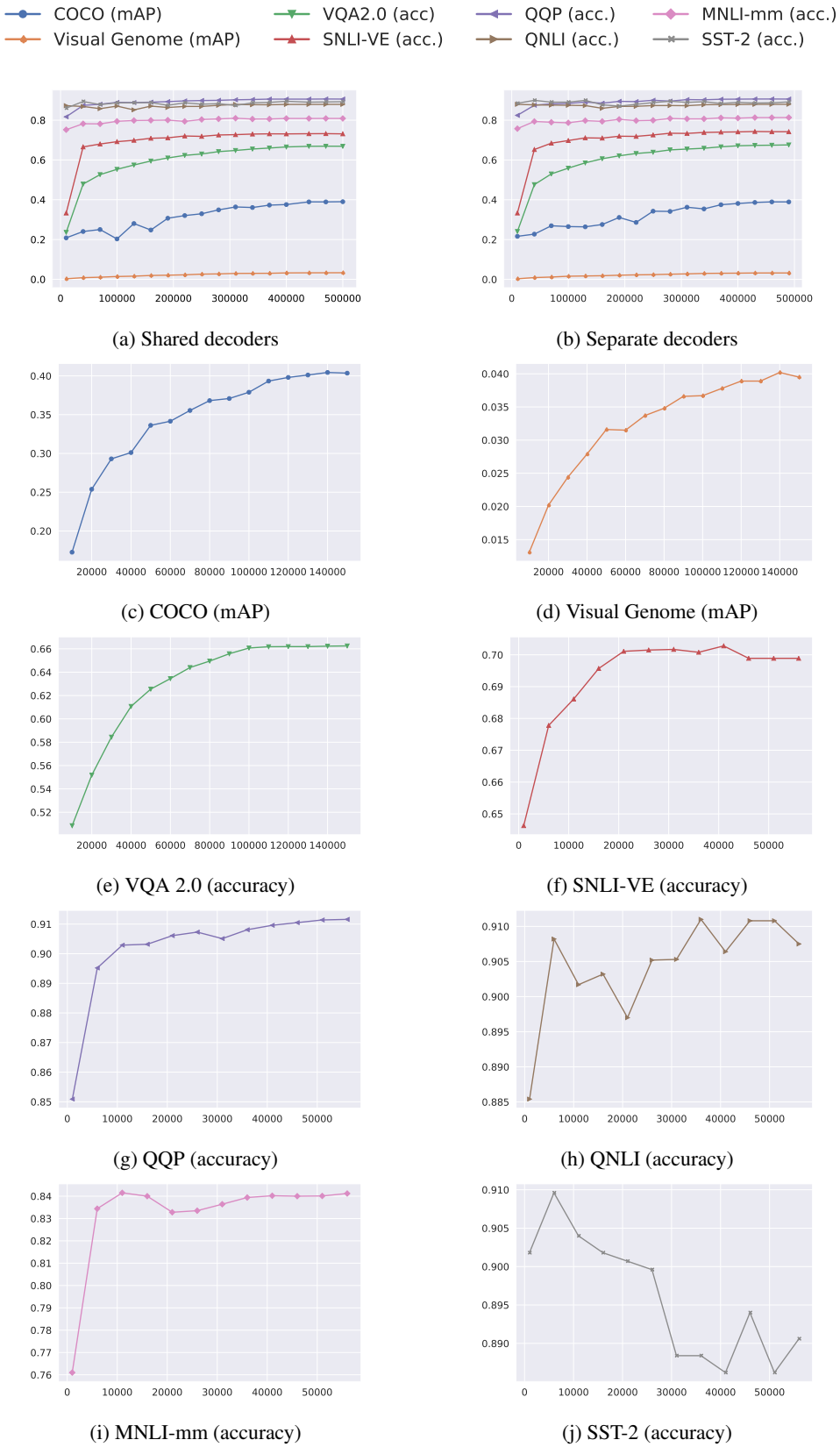
C. Learning curves

In Figure C.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.

D. More visualizations

Table D.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.



(a) Shared decoders

(b) Separate decoders

(c) COCO (mAP)

(d) Visual Genome (mAP)

(e) VQA 2.0 (accuracy)

(f) SNLI-VE (accuracy)

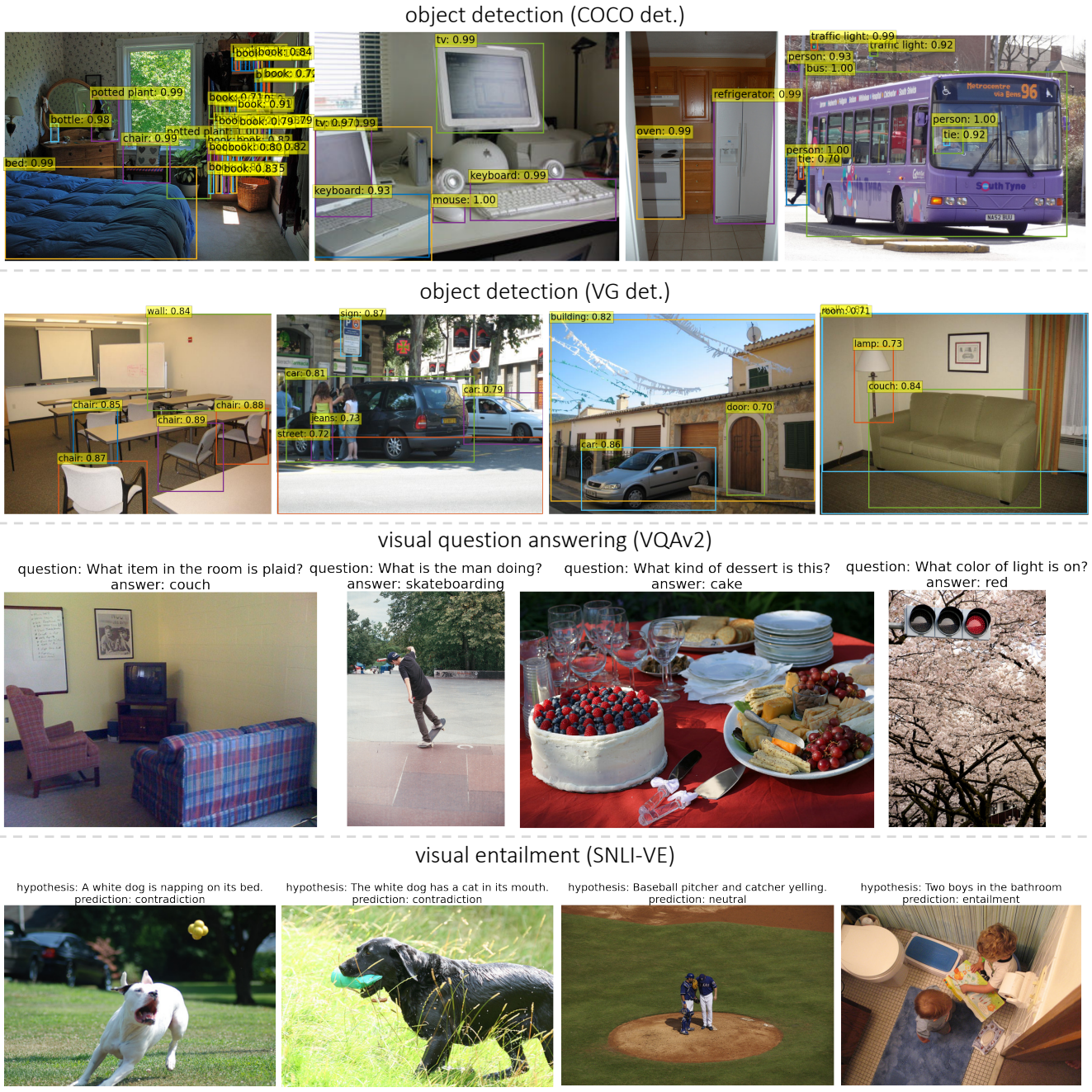
(g) QQP (accuracy)

(h) QNLI (accuracy)

(i) MNLI-mm (accuracy)

(j) SST-2 (accuracy)

Figure C.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.



QNLI

paragraph: As of that day, the new constitution heralding the Second Republic came into force.
question: What came into force after the new constitution was herald?
prediction: answerable

paragraph: For example, Joseph Haas was arrested for allegedly sending an email to the Lebanon, New Hampshire city councilors stating, "Wise up or die."
question: What year did the the case go before the supreme court?
prediction: cannot be answered

MNLI-mm

premise: Captain Victor Saracini and First Officer Michael Horrocks piloted the Boeing 767, which had seven flight attendants.
hypothesis: The Captain was Michael Horrocks and there were 4 flight attendants aboard.
prediction: contradiction

premise: They were promptly executed.
hypothesis: They were executed immediately upon capture.
prediction: neutral

QQP

question 1: Is there a reason why we should travel alone?
question 2: What are some reasons to travel alone?
prediction: equivalent

question 1: Why was the Roman Empire so successful?
question 2: What are some of the rarely known facts about the Roman Empire?
prediction: not equivalent

SST-2

paragraph: allows us to hope that nolan is poised to embark a major career as a commercial yet inventive filmmaker.
sentiment: positive

paragraph: in its best moments , resembles a bad high school production of grease , without benefit of song.
sentiment: negative

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