VirTex: Learning Visual Representations from Textual Annotations Supplementary Material

Karan Desai Justin Johnson
University of Michigan
{kdexd,justincj}@umich.edu

Appendix A. Additional Experiments

In this section, we describe additional implementation details about our experiments in Section 4. Our evaluation protocol is consistent with prior works on pretraining visual representations – we report differences where applicable.

A.1. Image Classification with Linear Models

PASCAL VOC: We use standard data augmentation on images from both `trainval` and `test` split – we resize the shorter edge to 256 pixels, and take a 224×224 center crop. We normalize images by ImageNet color (RGB mean = $[0.485, 0.456, 0.406]$, std = $[0.229, 0.224, 0.225]$).

Prior works [1–3] train per-class SVMs for $C \in [2^{-19}, 2^{-4}] \cup [10^{-7}, 10^{-2}]$ (26 values), and choose best SVM based on 3-fold cross-validation. In our initial evaluations, we observed that the best performing SVMs are typically trained with cost values $C \in \{0.01, 0.1, 1.0, 10.0\}$. Based on this observation, we only use these values for faster evaluation. For training SVMs, we use scikit-learn [4] with LIBLINEAR [5] backend, default parameters are:

```
LinearSVC(penalty='l2', dual=True,
max_iter=2000, tol=1e-4, class_weight={1: 2, -1: 1}, loss='squared_hinge').
```

ImageNet-1k: For data augmentation during training, we randomly crop 20–100% of the original image size, with a random aspect ratio in $(4/3, 3/4)$, resize to 224×224, apply random flip, and normalization by ImageNet color. During evaluation, we resize the shorter edge to 256 pixels and take a 224×224 center crop. We initialize the weights of the linear layer as $N(0.0, 0.01)$, and bias values as 0.

Note that we perform a small LR sweep separately for our VirTex model (ResNet-50 and $L = 1, H = 2048$), and ImageNet-supervised models. For Figure 4 best LR values for VirTex models is 0.3, and ImageNet-supervised models is 0.1.

Annotation Cost Efficiency: Here, we provide details on our cost estimates for different methods in Table 1. For labels and masks, we use estimates reported by COCO [6], and for captions we use estimates reported by `nocaps` [7], collected in a similar fashion as COCO.

- **Labels:** We consider total time of *Category Labeling* and *Instance Spotting* steps in [6] (∼30K hours). This estimate corresponds to 328K images – we scale it for COCO Captions `train2017` split (118K images).
- **Masks:** As reported in [6], it takes 22 worker hours for collecting 1000 instance segmentation masks. We use this estimate to compute time for ∼860K masks in COCO `train2017` split. The collection of masks is dependent on *Category Labeling* and *Instance Spotting*, we add the time for collecting labels in our total estimate.
- **Captions:** We use the median time per caption (39.2 seconds) as reported in [7] (∼151K captions) to estimate the cost of collecting (118K ×5) captions in COCO.

Data Efficiency: We train our ImageNet-supervised models on randomly sampled subsets of ImageNet (1%, 2% 5%, 10%, 20%, 50%). We sample training examples such that the class distribution remains close to 100% ImageNet. For VirTex models, we randomly sample 10%, 20%, 50%, and 100% of COCO Captions [8] – we do not use any class labels to enforce uniform class distribution. Note that *this may put ImageNet-supervised models at an advantage*.

We train our ImageNet-supervised models by following the *exact* setup used to train the publicly available ResNet-50 model in `torchvision`. We use SGD with momentum 0.9 and weight decay 10^{-4}. We use a batch size of 256, and perform distributed training across 8 GPUs (batch size 32 per GPU). We train for 90 epochs, with an initial learning rate 0.1, that is divided by 10 at epochs 30 and 60. We keep the number of training epochs fixed for models trained on smaller subsets of ImageNet (else they tend to overfit). For VirTex models, we scale training iterations according to the size of the sampled training set.

Comparison: ImageNet vs. Cropped COCO. Note that the ImageNet images mostly contain a single object (commonly called *iconic* images). On the other hand, COCO dataset contains ∼2.9 object classes and ∼5.7 instances per image. It may seem that VirTex requires fewer images than ImageNet-supervised models as they contain multiple objects per image. Here, we make an additional comparison

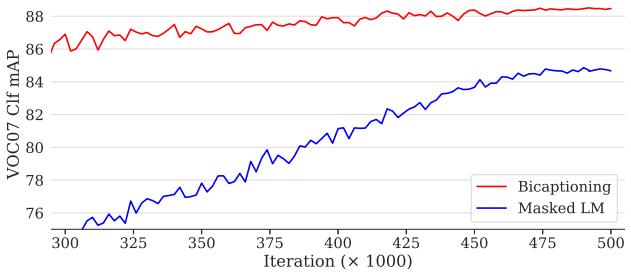

Figure 1: **Bicaptioning vs. Masked Language Modeling:** We compare VOC07 mAP of Bicaptioning and Masked LM pretraining tasks. We observe that Masked LM converges slower than Bicaptioning, indicating poor sample efficiency.

Backbone	VOC07	IN-1k	PASCAL VOC Detection		
	mAP	Top-1	AP^{bbox}_{all}	AP^{bbox}_{50}	AP^{bbox}_{75}
ResNet-50	88.3	53.2	55.2	81.2	60.8
ResNet-50 w2×	$88.5_{+0.2}$	$52.9_{-0.3}$	$56.6_{+1.4}$	$82.0_{+0.8}$	$62.8_{+2.0}$
ResNet-101	$88.7_{+0.4}$	$52.0_{-1.2}$	$57.9_{+2.7}$	$82.0_{+0.8}$	$63.6_{+2.8}$

Table 1: **Additional Evaluations for Backbone Ablations.** We compare VirTex models ($L = 1, H = 1024$) with different visual backbones. We observe that larger backbones generally improve downstream performance.

to control the varying image statistics between datasets.

Specifically, we crop objects from COCO images and create a dataset of 860K *iconic* images. We randomly expand bounding boxes on all edges by 0–30 pixels before cropping, to mimic ImageNet-like images. We train a ResNet-50 with same hyperparameters as ImageNet-supervised models, described above. It achieves **79.1** VOC07 mAP (vs. **88.7** VirTex). This shows that the data-efficiency of VirTex does not *entirely* stem from using scene images with multiple objects.

A.2. Ablations

Bicaptioning vs. Masked Language Modeling. In our pretraining task ablations (Section 4.2) we observed that Masked Language Modeling performs quite worse than all other pretraining tasks on downstream linear classification performance. This issue arises from the poor sample efficiency of Masked LM, discussed in Section 3

For more evidence, we inspect VOC07 mAP of Masked LM, validated periodically during training. In Figure 1, we compare this with VOC07 mAP of Bicaptioning. Both models use $L = 1, H = 2048$ textual heads. We find that Masked LM indeed converges slower than bicaptioning, as it receives weaker supervision per training caption – only corresponding to masked tokens. We believe that a longer training schedule may lead to MLM outperforming bicaptioning, based on its success in language pretraining [9].

```
_BASE_: "Base-RCNN-FPN.yaml"
INPUT:
  FORMAT: "RGB"
DATASETS:
  TRAIN: ("coco_2017_train",)
  TEST: ("coco_2017_val",)
MODEL:
  WEIGHTS: "Loaded externally"
  MASK_ON: True
  PIXEL_MEAN: [123.675, 116.280, 103.530]
  PIXEL_STD: [58.395, 57.120, 57.375]
  BACKBONE:
    FREEZE_AT: 0
  RESNETS:
    DEPTH: 50
    NORM: "SyncBN"
    STRIDE_IN_1X1: False
  FPN:
    NORM: "SyncBN"
SOLVER:
  IMS_PER_BATCH: 16
  BASE_LR: 0.02
  STEPS: (120000, 160000)
  MAX_ITER: 180000
TEST:
  PRECISE_BN:
    ENABLED: True
```

Table 2: **COCO Instance Segmentation:** Detectron2 config parameters that differ from base config file.

Additional Evaluation: Backbone Ablations. In our backbone ablations, we observed that larger visual backbones improve VOC07 classification performance. However, the performance trend for ImageNet-1k linear classification is opposite. We think this is an optimization issue – the hyperparameters chosen for ResNet-50 may not be optimal for other backbones. To verify our claims, we evaluate these models on PASCAL VOC object detection.

In Table 1, we observe that the performance trends of PASCAL VOC object detection match with VOC07 classification. Hence, we conclude that using larger visual backbones can improve downstream performance.

A.3. Fine-tuning Tasks for Transfer

We described the main details for downstream fine-tuning tasks in Section 4.3. We provide config files in Detectron2 [10] format to exactly replicate our downstream fine-tuning setup for COCO (Table 2), PASCAL VOC (Table 3), LVIS (Table 4). We apply modified hyperparameters on top of base config files available at:
github.com/facebookresearch/detectron2 @ b267c6

iNaturalist 2018 Fine-grained Classification: We use data augmentation and weight initialization same as ImageNet-1k linear classification (Section A.1). Despite a long-tailed distribution like LVIS, we do not perform class balanced resampling, following the evaluation setup of MoCo [11].

LVIS v0.5 Instance Segmentation: We already evaluated VirTex and baseline methods on LVIS Instance Segmenta-

```
_BASE_: "Base-RCNN-C4.yaml"
INPUT:
  FORMAT: "RGB"
  MIN_SIZE_TRAIN: (480, 512, 544, 576, 608, 640,
                   672, 704, 736, 768, 800)
DATASETS:
  TRAIN:("voc_2007_trainval","voc_2012_trainval")
  TEST: ("voc_2007_test",)
MODEL:
  MASK_ON: False
  WEIGHTS: "Loaded externally"
  PIXEL_MEAN: [123.675, 116.280, 103.530]
  PIXEL_STD: [58.395, 57.120, 57.375]
  BACKBONE:
    FREEZE_AT: 0
  RESNETS:
    DEPTH: 50
    NORM: "SyncBN"
    STRIDE_IN_1X1: False
  FPN:
    NORM: "SyncBN"
  ROI_HEADS:
    NUM_CLASSES: 20
SOLVER:
  IMS_PER_BATCH: 16
  BASE_LR: 0.02
  STEPS: (18000, 22000)
  MAX_ITER: 24000
  WARMUP_ITERS: 100
TEST:
  PRECISE_BN:
    ENABLED: True
```

Table 3: **PASCAL VOC Object Detection:** Detectron2 config parameters that differ from base config file.

tion task using LVIS `v1.0 train` and `val` splits. One of our baselines, MoCo, conducted this evaluation using `LVIS v0.5` splits. For completeness, we report additional results on LVIS `v0.5` split. The main changes in config (Table 4) following original LVIS `v0.5` baselines are: `NUM_CLASSES: 1230` and `SCORE_THRESHOLD_TEST: 0.0`

Results are shown in Table 5. We observe the VirTex significantly outperforms all baseline methods on LVIS `v0.5` split, similar to evaluation on LVIS `v1.0` split.

A.4. Selecting Best Checkpoint by VOC07 mAP

As described in Section 3, we observed that image captioning performance has an imprecise correlation with performance on downstream vision tasks. Hence, we select our best checkpoint based on VOC07 classification mAP.

In Figure 2, we compare validation metrics of our best VirTex model (ResNet-50, $L = 1, H = 2048$). We observe the trends of VOC07 mAP and CIDEr [12] score of the forward transformer decoder. We observe that an improvement in captioning performance indicates an improvement in downstream performance. However these are not strongly correlated – the best performing checkpoints according to these metrics occur at different iterations: 492K according to VOC07 mAP (**88.7**), and 480K according to CIDEr (**92.4**). Hence, we select the best checkpoint based on PASCAL VOC linear classification performance. We use this task as a representative downstream vision task for evaluation due to its speed and simplicity.

```
_BASE_: "Base-RCNN-FPN.yaml"
INPUT:
  FORMAT: "RGB"
DATASETS:
  TRAIN: ("lvis_v1.0_train",)
  TEST: ("lvis_v1.0_val",)
DATALOADER:
  SAMPLER_TRAIN: "RepeatFactorTrainingSampler"
  REPEAT_THRESHOLD: 0.001
MODEL:
  WEIGHTS: "Loaded externally"
  MASK_ON: True
  PIXEL_MEAN: [123.675, 116.280, 103.530]
  PIXEL_STD: [58.395, 57.120, 57.375]
  BACKBONE:
    FREEZE_AT: 0
  RESNETS:
    DEPTH: 50
    NORM: "SyncBN"  # For IN-sup: "FrozenBN"
    STRIDE_IN_1X1: False
  FPN:
    NORM: "SyncBN"  # For IN-sup: ""
  ROI_HEADS:
    NUM_CLASSES: 1230
    SCORE_THRESH_TEST: 0.0
SOLVER:
  IMS_PER_BATCH: 16
  BASE_LR: 0.02
  STEPS: (120000, 160000)
  MAX_ITER: 180000
TEST:
  DETECTIONS_PER_IMAGE: 300
  PRECISE_BN:
    ENABLED: True
```

Table 4: **LVIS Instance Segmentation:** Detectron2 config parameters that differ from base config file.

Method	Pretrain Images	LVIS v0.5 Instance Segmentation		
		AP^{bbox}_{all}	AP^{bbox}_{50}	AP^{bbox}_{75}
1) Random Init		22.5	34.8	23.8
2) IN-sup	1.28M	24.5	38.0	26.1
3) IN-sup-50%	640K	23.7 –0.8	36.7 –1.3	25.1 –1.0
4) IN-sup-10%	128K	20.5 –4.0	32.8 –6.2	21.7 –5.2
5) MoCo-IN	1.28M	24.1 –0.4	37.4 –0.6	25.5 –0.6
6) MoCo-COCO	118K	23.1 –1.4	35.3 –2.7	24.9 –1.2
7) VirTex	118K	25.4 +0.9	39.0 +1.0	26.9 +0.8

Table 5: **Downstream Evaluation: LVIS v0.5 Instance Segmentation.** We compare VirTex with different pretraining methods for LVIS `v0.5` Instance Segmentation. All methods use Mask R-CNN with ResNet-50-FPN backbone. Performance gaps with IN-sup are shown on the side. The trends are similar to LVIS `v1.0` (Table 3) – VirTex significantly outperforms all baseline methods.

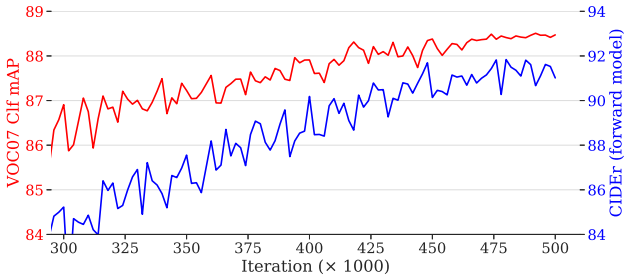

Figure 2: **Validation metrics: VOC07 mAP and CIDEr.** We compare VOC07 mAP and CIDEr score of VirTex (ResNet-50, $L = 1, H = 2048$) model. We observe that captioning performance has a positive, yet imprecise correlation with downstream performance on vision tasks.

Appendix B. Decoder Attention Visualizations for Caption Predictions

In Figure 3 and Figure 4, we show more qualitative examples showing decoder attention weights overlaid on input images. All captions are decoded from $L = 1, H = 512$ VirTex model using beam search. We normalize the attention masks to $[0, 1]$ to improve contrast for better visibility.

References

[1] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra, "Scaling and benchmarking self-supervised visual representation learning," in *CVPR*, 2019.

[2] Ishan Misra and Laurens van der Maaten, "Self-supervised learning of pretext-invariant representations," in *CVPR*, 2020.

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *NeurIPS*, 2020.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[5] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, "LIBLINEAR: A library for large linear classification," *Journal of machine learning research*, 2008.

[6] 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 *ECCV*, 2014.

[7] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson, "nocaps: novel object captioning at scale," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8948–8957, 2019.

[8] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick, "Microsoft COCO captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*, 2015.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[10] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick, "Detectron2." https://github.com/facebookresearch/detectron2, 2019.

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020.

[12] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh, "CIDEr: Consensus-based image description evaluation," in *CVPR*, 2015.

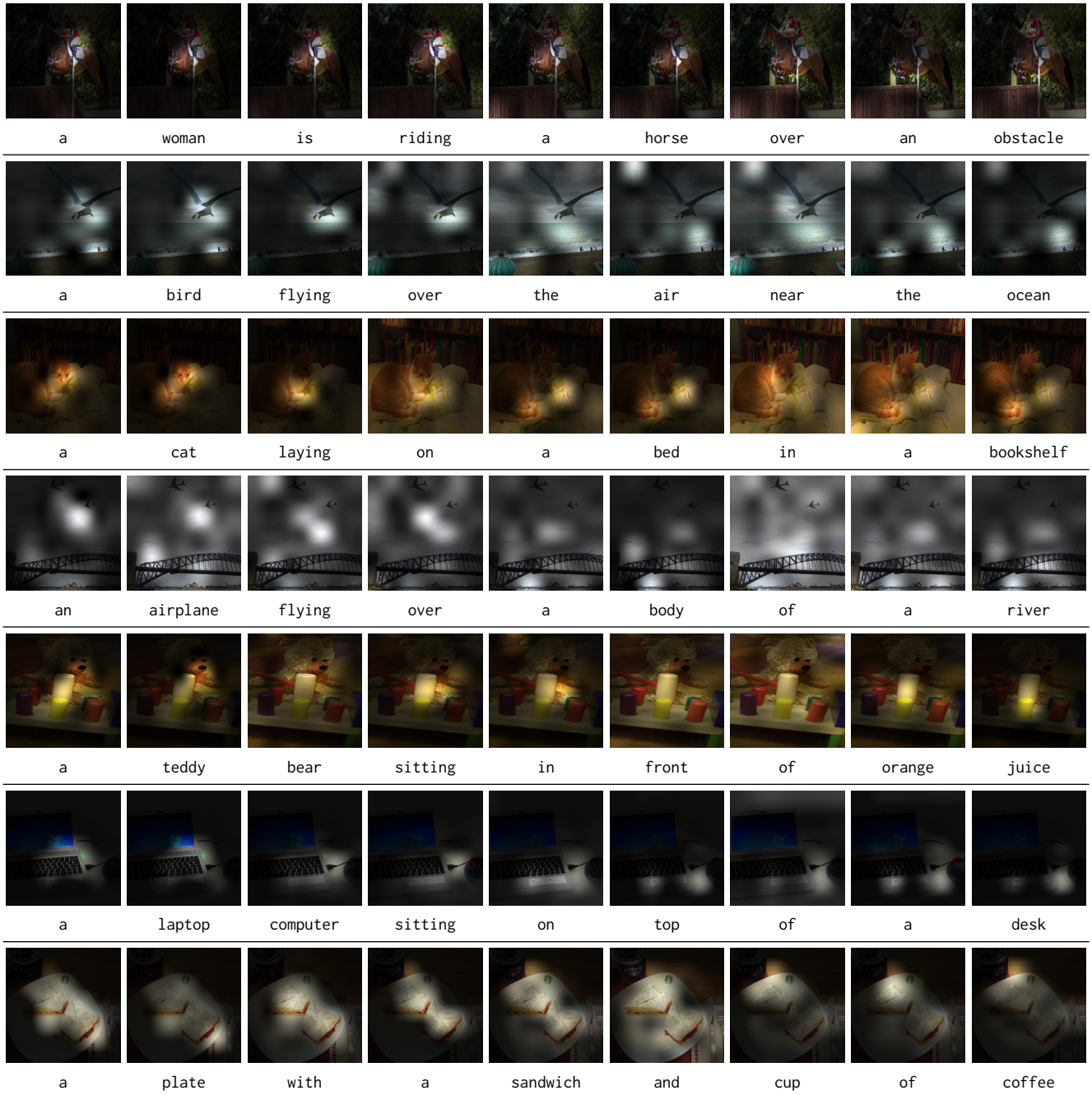

Figure 3: Attention visualizations per time step for predicted caption. We decode captions from the forward transformer of $L = 1, H = 512$ VirTex model using beam search.

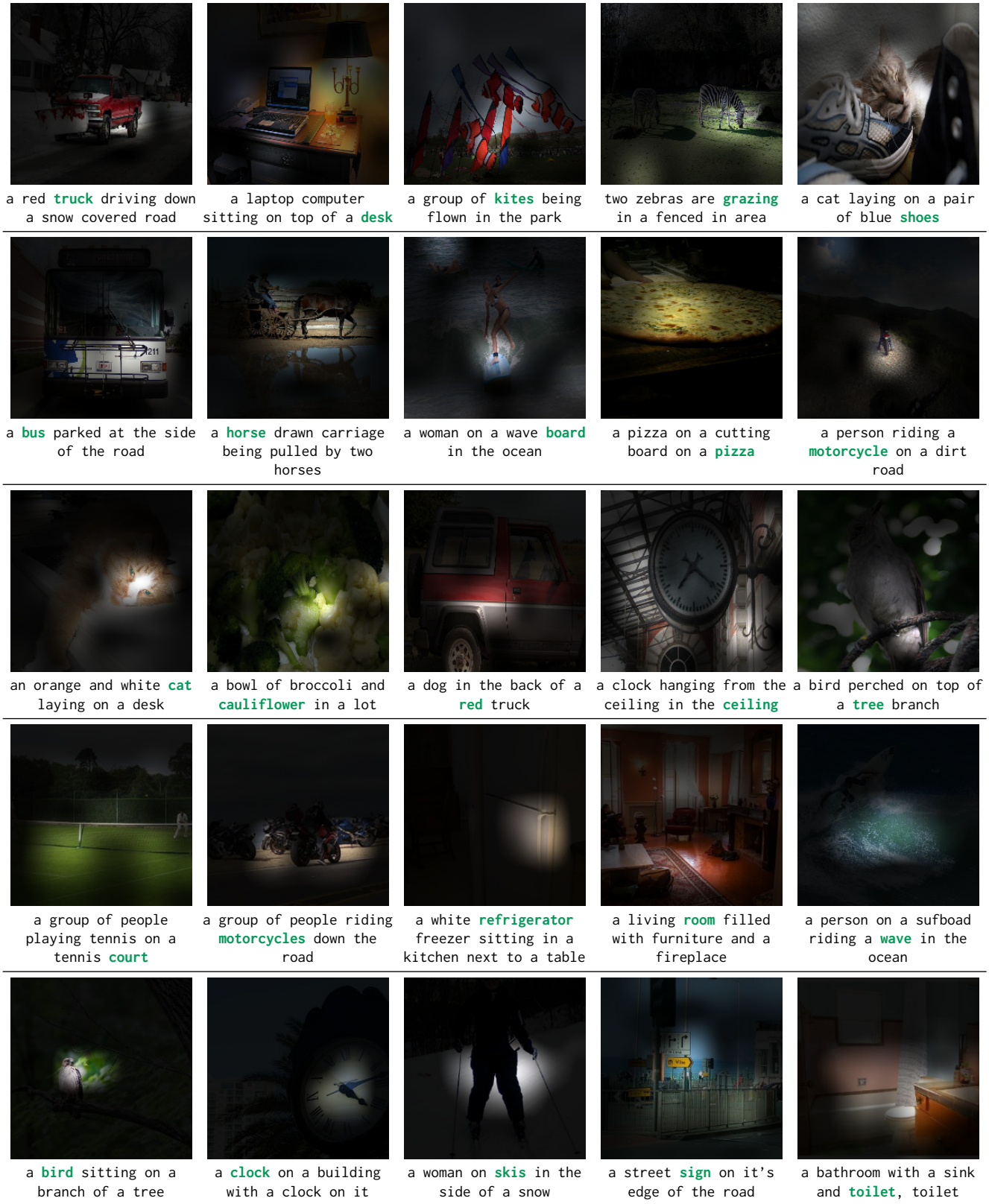

Figure 4: We decode captions from the forward transformer of $L = 1, H = 512$ VirTex model using beam search. For the highlighted word, we visualize the decoder attention weights overlaid on the input image.