**EK Benchmark May Not Be Correct (R1).** We do not derive the EK ground truth from noisy narrations. The EK dataset contains 1) $(noun, verb)$ action tuple labels, 2) bounding box annotations of *active* objects, and 3) a list of $nouns$ associated with an action (e.g., ['pan', 'mushrooms'] for "put mushrooms in the pan"). We construct the ground truth by finding frames where the object category of a bounding box matches the $noun$ of either 1) an action tuple or 2) one of the $nouns$ from a $noun$ list. We manually verified our annotations and confirmed that they were error-free.

**Claims of Not Training COBE on EK (R1, R2, R4).** We acknowledge that even though we do not directly train COBE on EK, we use a detector trained on EK to pseudo-label the frames of HowTo100M, which might ease the transfer to EK. We will revise our former claims to make this point clear. Also, note that we do *not* start from a detector pre-trained on EK when training on HowTo100M_BB. To expand our evaluation beyond EK, we also manually labeled a test set of $9K$ HowTo100M frames (using a disjoint set from HowTo100M_BB) with $171$ unique $(noun, context)$ tuples, where $context$ can be one of: $noun, verb, adjective$ or $adverb$. On this new set, COBE achieves $17.5$ mAP, while Tuple Faster R-CNN yields $16.1$ mAP. We will release all benchmark annotations upon publication of the paper.

**Additional Baselines (R1, R2, R3, R4).** As requested by R3, and R4, we use S3D trained on HowTo100M [Miech et al., CVPR20] with Faster R-CNN to construct a baseline for our task (as for the other baselines in Table 1 of our draft). While this baseline performs better than other action recognition baselines, its mAP accuracy is $7.9$ and $10.6$ lower than that achieved by COBE on the EK_H evaluation set for $(noun, noun)$, and $(noun, verb)$, respectively. In response to R1's question, we note that these results indicate that classification systems such as those of Miech et al. cannot be easily adapted to our problem, and that a specialized design such as COBE might be needed for better solving this task. Also, as suggested by R3, we replaced CTRL, with a word2vec embedding, which produced mAP accuracies of $15.8$ and $21.9$ compared to the $16.9$ and $24.7$ of COBE w/ CTRL. We also tried the weighted averaging scheme suggested by R3, which yielded $14.9$ and $22.1$ in mAP. Lastly, per R2's request, we implemented a baseline that "matches object detection outputs to noun tokens," which achieved a $6.7$ mAP on the EK_H (compared to the $16.9$ mAP of COBE).

**Object Detection Results (R2, R3).** As requested, we ran object detection experiments on $124K$ frames of EK ($180$ object categories) by comparing COBE to a Faster R-CNN trained on HowTo100M_BB for object detection. Both methods share the same architecture (except for the contextualized object branch). COBE outperforms this baseline by $1.4$ mAP. We also note that pre-training COBE on HowTo100M_BB and then finetuning it on EK outperforms Faster R-CNN only trained on EK by $10.1$ mAP. This highlights the benefit of pretraining on HowTo100M_BB.

**Limitations of COBE (R3, R4).** R3 and R4 correctly point out that COBE relies on a predefined set of noun-centric object categories. However, because of the supervision from narration, our detector can recognize fine-grained contextual cues beyond the coarse categorical labels. For example, a standard object detector can predict that there is a "tomato" in an image, whereas COBE can predict that there is "a yellow tomato" or "a chopped tomato." R4 rightly points out that unseen tuple categories may still occur visually in the training frames even though they are not mentioned in the text. However, because COBE is supervised by textual captions, it would *not* be trained to predict the categories of those "unmentioned" tuples. Thus, in those cases, COBE is indeed inferring new tuple concepts. We fully agree with R3 and R4 that it would be great to eliminate the reliance on bounding box labels. However, this is challenging, particularly because HowTo100M captions are noisy. We intend to tackle this problem in our future work.

**Related Work (R4).** ViLBERT is better suited for tasks that require multi-modal inputs (e.g. VQA, VCR). Instead, COBE uses captions to supervise a visual detection model during training, but then operates on video frames alone during inference. We thank R4 for highlighting these related models, which we will cite and discuss in our final paper.

**EK Action Recognition Results (R2).** Unlike action recognition models, which are trained on manually annotated action labels, COBE is trained on noisy narrations. Thus, such a comparison would not be properly fair. However, we agree with R2 that such experiments would be useful and we will attempt to include them in our final draft.

**Technical Novelty (R1, R2, R3).** We acknowledge that our work offers limited contributions in terms of model design. Our main technical contribution is our large-scale training framework which leverages noisy narrations for learning object detectors that generalize to novel states. This is in contrast to prior work [43, 44] that focuses on noun-centric object detection, and that relies on smaller datasets with manually annotated text descriptions. As noted by R4, our idea is simple, and effective, and it demonstrates a novel object detection application of modern language models.

**Experimental Questions (R2, R4).** 1) Prior to our evaluation, we verified that our evaluation set does *not* contain frames with multiple tuple categories. However, we agree with R2 that for instances with multiple valid tuples, using a softmax would be suboptimal. To address this issue, we would use a sigmoid instead of a softmax in our formulation in L210. 2) The two tasks are indeed separated to include specialized baselines. 3) We considered a 3D backbone, but it was too costly to train it on HowTo100M_BB. 4) Misalignment between captions and videos was not an issue because we used only the most confident detections to construct HowTo100M_BB (see L118-L120). 5) We appreciate R4's advice of using global features for capturing more context: we will try it. 6) We confirm that: a) Faster R-CNN is trained to predict the classes from EK, and b) COBE is trained end-to-end.