Weakly-Supervised Generation and Grounding of Visual Descriptions with Conditional Generative Models - Supplementary Material

Effrosyni Mavroudi and René Vidal
Mathematical Institute for Data Science, Dept. of Biomedical Engineering, Johns Hopkins University
{emavrou1, rvidal}@jhu.edu

This Appendix provides (a) additional technical details about training and inference (Section A), (b) implementation details per dataset (Section B), and (c) qualitative results (Section C), to supplement our main paper. These are not included in the main paper due to space constraints.

A. Additional technical details

In this Section we provide additional technical details that were not included in the main paper due to space constraints:

- In Section A.1, we add more details about our training approach.
- In Section A.2, we describe our approximate inference approach.

A.1. Training

To optimize the final hybrid objective (Eq. 10) using Stochastic Gradient Descent, we approximate the expectation for the reconstruction loss term of the CVAE loss (Eq. 11) using a Monte-Carlo estimator, with S region samples $\mathbf{z}_t^{(s)}$ drawn from $q_\phi(\mathbf{z}_t \mid Y, R, I)$, where S is a hyperparameter:

$$\mathop{\mathbb{E}}_{\mathbf{z}_t \sim q_\phi}\left[-\log p_\theta(\mathbf{y}_t^{(n)} \mid \mathbf{y}_{<t}^{(n)}, \mathbf{z}_t, R^{(n)})\right] \approx -\frac{1}{S}\sum_{s=1}^{S}\log p_\theta(\mathbf{y}_t^{(n)} \mid \mathbf{y}_{<t}^{(n)}, \mathbf{z}_t^{(s)}, R^{(n)}) \quad (1)$$

To backpropagate gradients of the reconstruction loss term all the way to the parameters ϕ of the inference model through the sampling, as our latent variables are discrete, we sample from the Gumbel-Softmax [1, 8] continuous approximation of the categorical distribution q_ϕ with temperature τ. This distribution converges to one-hot samples from the categorical distribution when $\tau \to 0$ and it allows us to use the reparameterization trick [4] to compute (biased) gradients w.r.t. parameters ϕ.

A.2. Approximate inference

Visual Object Grounding. Given an input image or video and a ground-truth sentence Y, we address the VOG task by inferring the latent word-to-region alignment for each word of the sentence using the approximate posterior word-to-region alignment distribution (CVAE-q) and using that to select a region proposal:

$$\widehat{\mathbf{b}}_t = \mathbf{r}_j, \text{ where } j = \mathop{\text{argmax}}_{i \in \{1,\dots,M\}} q_\phi(z_{t,i} = 1 | Y, R, I). \quad (2)$$

We also experimented with using the prior word-to-region alignment distribution for grounding (CVAE-p). Although this distribution suffers from the same limitation as soft-attention, namely it does not take into account the word being grounded, our experimental results suggest that it outperforms soft-attention:

$$\widehat{\mathbf{b}}_t = \mathbf{r}_j, \text{ where } j = \mathop{\text{argmax}}_{i \in \{1,\dots,M\}} p_\theta(z_{t,i} = 1 | \mathbf{y}_{<t}, R, I). \quad (3)$$

Both grounding approaches have the same computational complexity as popular grounding methods [17, 18]: they require passing the sentence through a (Bi)LSTM and applying an attention mechanism (similarity function) over regions for each word.

We would like to clarify that our model assumes a single latent region for each groundable word both for images and videos. Therefore, given an input video and a textual description, each groundable word is localized with a bounding box in a potentially different frame of the video. However, we would often like to ground words in particular frames of the video. To do this, we use a heuristic, i.e., we choose the region at frame l with maximum q-attention coefficient (or p-attention coefficient):

$$\widehat{\mathbf{b}}_{t,l} = \mathbf{r}_j, \text{ where } j = \mathop{\text{argmax}}_{i \in \mathcal{J}_l} q_\phi(z_{t,i} = 1 | Y, R, I), \quad (4)$$

where $\mathcal{J}_l$ is the set of region indices extracted from frame l.

Grounded Visual Description. For the task of GVD, we follow a two-stage approach: first we generate a sentence and then we ground the generated words. Similar to Zhou et al. [17], we perform greedy decoding for sentence generation, i.e. we predict a word y_t^* at each timestep t and feed it as input to the next timestep. In particular, each next word can be predicted by using the marginal word distribution:

$$\widehat{\mathbf{y}}_t = \underset{\mathbf{y}_t}{\operatorname{argmax}} \underset{\mathbf{z}_t \sim p_\theta(z_t|\mathbf{y}^*_{<t}, R, I)}{\mathbb{E}} p_\theta(\mathbf{y}_t|\mathbf{z}_t, \mathbf{y}^*_{<t}, R, I), \quad (5)$$

The marginal word distribution can be approximated via Monte Carlo sampling:

$$\widehat{\mathbf{y}}_t = \underset{\mathbf{y}_t}{\operatorname{argmax}} \frac{1}{K} \sum p_\theta(\mathbf{y}_t|\mathbf{z}_t^{(k)}, \mathbf{y}^*_{<t}, R, I), \quad (6)$$

where $\left\{\mathbf{z}_t^{(k)}\right\}_{k=1}^{K}$ are K samples drawn according to $p_\theta(z_t|\mathbf{y}^*_{<t}, R, I)$. However, doing so is computationally expensive. Instead, we feed the expected value of $\mathbf{z}_t$:

$$\widehat{\mathbf{y}}_t = \underset{\mathbf{y}}{\operatorname{argmax}}\, p_\theta(\mathbf{y}_t| \underset{\mathbf{z}_t \sim p_\theta}{\mathbb{E}} [\mathbf{z}_t], \mathbf{y}^*_{<t}, R, I). \quad (7)$$

Observe that $\mathbb{E}_{\mathbf{z}_t \sim p_\theta}[\mathbf{z}_t] = a_\theta(\mathbf{s}_t, X) \in \mathbb{R}^M$, i.e. the expected value of $\mathbf{z}_t$ is equal to the attention coefficients computed by the p-attention network. Therefore, Eq. 7 can be rewritten as:

$$\widehat{\mathbf{y}}_t = \operatorname{argmax}\left[\operatorname{softmax}\left(\operatorname{MLP}_\theta\left[\sum_{i=1}^{M} a_\theta^{(i)}(\mathbf{s}_t, X)\mathbf{x}_i; \mathbf{s}_t\right]\right)\right]. \quad (8)$$

Note that $p_\theta(\mathbf{y}_t| \mathbb{E}_{\mathbf{z}_t \sim p_\theta}[\mathbf{z}_t], \mathbf{y}^*_{<t}, R, I)$ is a first-order Taylor approximation of the expectation in Eq. 5. If we choose to use a single layer MLP for word prediction:

$$g_\theta(\mathbf{s}_t, \mathbf{z}_t, X) = \operatorname{softmax}\left(W_c\left[\sum_{i=1}^{M} z_{t,i}\mathbf{x}_i; \mathbf{s}_t\right]\right), \quad (9)$$

$$\mathbf{y}_t \mid \mathbf{y}_{<t}, \mathbf{z}_t, R, I \sim \operatorname{Cat}(g_\theta(\mathbf{s}_t, \mathbf{z}_t, X)), \quad (10)$$

then $p_\theta(\mathbf{y}_t| \mathbb{E}_{\mathbf{z}_t \sim p_\theta}[\mathbf{z}_t], \mathbf{y}^*_{<t}, R, I)$ is also valid lower bound of $\mathbb{E}_{\mathbf{z}_t}\, p_\theta(\mathbf{y}_t|\mathbf{z}_t, \mathbf{y}^*_{<t}, R, I)$, since the single layer MLP is a convex function (composed of a linear mapping and the softmax function) of $\mathbf{z}_t$ and we can apply Jensen's inequality. Using the expected value of $\mathbf{z}_t$ serves as a shortcut to avoid sampling, thus retaining the same computational complexity as discriminative encoder-decoder captioning methods.

Given the generated sentence $\hat{Y}$, we can use the prior (Eq. 3) or approximate posterior (Eq. 2) word-to-region alignment distributions to ground the generated words.

B. Additional experimental details

In this section we provide additional details about the models and the experimental setup of the experiments reported in the main paper. Note that we provide detailed implementation details for each dataset. Most of our modules and hyperparameters are either following the setup GVD [17] or are shared between datasets.

- In Section B.1, we describe the implementation details and experimental setup for our experiments on the Flickr30k Entities dataset. We also report the standard deviations for our reported results on the Flickr30k Entities test set, compare our GVD-CVAE with baselines on the Flickr30k Entities test set, report the standard deviations for our reported results on the Flickr30k Entities test set, and plot learning curves in Fig. 1.
- In Section B.2, we describe the implementation details and experimental setup for our experiments on the ActivityNet Entities dataset. We also provide standard deviations for our reported results on the ActivityNet Entities validation set.
- In Section B.3, we describe the implementation details and experimental setup for our experiments on the YouCook2 dataset. We also provide an extended version of Table 6 with additional metrics and method details (Table 3), and results on the validation set (Table 4).
- In Section B.4, we discuss average runtime, computing infrastructure and the open-source codebases and public datasets that we used in our experiments.

B.1. Additional experimental details for Flickr30k Entities

Implementation and experimental setup for the results reported in Table 4 of the main paper (GVD-CVAE):

Inputs. We use the same region proposals and features as Zhou et al. [17]. For each image, we use a Faster R-CNN [9] detector with ResNext-101 [15] backbone pretrained on Visual Genome [5] to obtain region proposals. In particular, we retain the top 100 region proposals per frame, based on their detection confidence score. Each region is described by a 2048-dimensional feature vector extracted from the $fc6$ layer of the ResNext-101. Same as GVD [17], we also use a global feature vector of size 2048 describing the image to be captioned. We use a vocabulary of 8639 words including `UNK` (the symbol for rare words not included in the vocabulary) and `EOS` (end of sentence special symbol). Words are embedded to a 512-dimensional vector using randomly initialized embeddings, trained from scratch, same as in GVD [17].

Model. The pre-extracted region features, image convolutional features and global image feature are transformed into $\mathbf{x}$, F and $\mathbf{v}$ by our trainable encoder (which mirrors the encoder of GVD [17]). In particular, the region embedding consists of the concatenation of: a linear projection of the $fc6$ region feature (initialized with the $fc7$ layer weights of the object detector), a 300-dimensional trainable embedding of the 4-dimensional position of the bounding box coordinates, and a 481-dimensional vector of object classification scores obtained by applying a trainable object classification layer on top of the $fc7$ feature. Note that these object classification scores are also used in our inference model when $\gamma = 1$. [1] After normalizing these 3 components with layernorm, they are concatenated and passed through a linear projection that projects to a lower-dimensional space of dimensionality 1024. This serves as our *grounding-aware region embedding* [17] $\mathbf{x}_i$. Similarly, the convolutional features and global image feature are projected with two linear transformation layers to a lower-dimensional space of dimensionality 1024, yielding F and $\mathbf{v}$, respectively. Our decoder has at its core a two layer (hierarchical) LSTM of hidden size 1024 (Eq. 5). The convolutional attention $f_\theta(\cdot,\cdot)$ is a 'concat' attention mechanism of attention size 512, which takes in the convolutional feature map F and determines by the hidden state $\mathbf{u}_t$ how significant each feature map column should contribute to generate a word ($f_\theta^{(l)}(\mathbf{u}_t, F)$). The same holds of the region attention $k_\theta(\cdot,\cdot)$. Our p-attention network is an 'concat' attention mechanism with attention size 512. Our full inference model consists of a Bi-LSTM with hidden size 1024 and a q-attention network with dot-product attention mechanism ($\alpha_\phi^{(i)}(\mathbf{h}_t, \mathbf{x}) \propto \mathbf{h}_t^T W \mathbf{x}_i$). It also uses information about object classes from an external dataset ($\gamma = 1$).

Training. We train our model for 40 epochs with the Adam [3] optimizer, having an initial learning rate of $2e-4$, decayed by a factor 0.8 every 3 epochs. Our batch size is 40 images and $S = 10$, $\tau = 0.8$. Note that we start training with $\lambda = 0$ for 20 epochs and then add the ELBO loss and jointly optimize the cross-entropy and ELBO losses ($\lambda = 0.5$). For annealing β, we use the PI-Controller [11], with $Ki = -0.0001$, $exp_{kl} = 0.06$, $Kp = 0.01$. Hyperparameters were either borrowed by GVD or were chosen based on Box accuracy on the validation set. We apply dropout 0.5 on fully-connected layers. All layers are trained from scratch, except for the backbones yielding the initial region and image features. Ground-truth captions are truncated to 20 words during training and testing.

[1] We would like to emphasize that this object class knowledge from pretrained object detectors cannot substitute the full supervision of annotated bounding boxes per groundable word, since many words do not belong in the classes of the object detector, and grounding a word is a different task than object detection; we need to localize the referred entity instead of all entities belonging to an object class.

Note that to fairly compare with methods that use Reinforcement Learning for finetuning the captioning model, we also finetuned our decoder and prior networks with SCST using CIDEr as the reward (GVD-CVAE-SCST). Since the goal of this experiment was to show that our decoder can be finetuned with RL to match the performance of SoTA models in captioning, we chose to finetune a simpler model (hence the small reduction in weakly-supervised grounding from 33.8 to 31.6 - Table 4 in our main paper, last 2 rows). In particular, we used a simpler model that was trained with our hybrid loss until epoch 38 (with an LSTM inference network and $\gamma = 0$) and then finetuned with SCST until epoch 60, with learning rate $5e-5$ and batch size 48.

Evaluation. We evaluate our model (the checkpoint at the end of training) on weakly-supervised object grounding and grounded captioning on the validation and testing sets. We use the GVD metrics and evaluation scripts for evaluating: captioning and grounding [2]. This yields the reported result in Table 4, row 11 (GVD-CVAE).

B.2. Additional experimental details for ActivityNet Entities

Implementation and experimental setup for the results reported in Table 5 of the main paper.

Inputs. We use the same region proposals and features as Zhou et al. [17]. For each frame, we use a Faster R-CNN [9] detector with ResNext-101 [15] backbone pretrained on Visual Genome [5] to obtain region proposals. In particular, we retain the top 100 region proposals per frame, based on their detection confidence score. Each region is described by a 2048-dimensional feature vector extracted from the $fc6$ layer of the ResNext-101. We also combine that region feature with a 300-dimensional trainable embedding of the bounding box coordinates (including the normalized frame index) and a 432-dimensional vector of object classification scores (yielding the *grounding-aware region encoding* of GVD [17]). We also use a global feature vector of size 3072 describing the video segment to be captioned, which is obtained by averaging the temporal sequence of frame-wise appearance and motion features from [17]. Following GVD [17] the global feature vector is augmented with a 50-dimensional embedding of the segment positional information (i.e., total number of segments, segment index, start time and end time). We use a vocabulary of 4905 words including `UNK` (the symbol for rare words not included in the vocabulary) and `EOS` (end of sentence special symbol). Words are embedded to a 512-dimensional vector using randomly initialized embeddings, trained from scratch, same

[2] https://github.com/facebookresearch/grounded-video-description/blob/44411533ea967244867a6b186a9b5cebba476015/eval_grd_flickr30k_entities.py

Table 1. Results on the Flickr30k Entities test set. Supplements Table 4 of the main paper with standard deviations and grounding results from CVAE-p (the prior distribution is used for captioning in both GVD-CVAE-p and GVD-CVAE-q).

		VOG	GVD					
			Captioning				Grounding	
	Feat	Acc	B@4	M	C	S	$F1_{all}$	$F1_{loc}$
GVD [17]	G	21.4	26.9	22.1	60.1	16.1	3.88	11.7
GVD-Grd [17]	G	25.5	26.9	22.1	60.1	16.1	3.88	11.7
GVD-CVAE-p	G	30.4 ± 1.0	24.0 ± 0.6	21.3 ± 0.1	55.3 ± 1.3	15.7 ± 12.1	6.4 ± 0.2	18.1 ± 0.6
GVD-CVAE-q	G	33.7 ± 0.4	24.0 ± 0.6	21.3 ± 0.1	55.3 ± 1.3	15.7 ± 12.1	6.70 ± 0.5	19.2 ± 1.2

Table 2. Results on the ActivityNet Entities validation set. Supplements Table 5 of the main paper with standard deviations for our GVD-CVAE.

	VOG	GVD					
		Captioning				Grounding	
	Acc	B@4	M	C	S	$F1_{all}$	$F1_{loc}$
GVD [17]	14.9	2.28	10.9	45.6	15.0	3.7	12.7
GVD-Grd [17]	21.3	2.28	10.9	45.6	15.0	3.7	12.7
GVD-CVAE	23.9 ± 0.5	1.90 ± 0.03	10.4 ± 0.03	41.8 ± 0.4	13.3 ± 0.1	5.8 ± 0.4	21.7 ± 1.6

as in GVD [17]. In summary, we use the same inputs as the compared methods: GVD [17] and Cyclical [7].

Model. Following GVD [17], the pre-extracted region features and global video feature are transformed into $\mathbf{x}$ and $\mathbf{v}$ by our trainable encoder, i.e. a linear mapping to a lower-dimensional space of dimensionality 1024. Similarly, frame-wise global features F are encoded with a Gated Recurrent Unit (GRU) with hidden size 1024. Our decoder has at its core a two layer (hierarchical) LSTM of hidden size 1024. The temporal attention is an 'concat' mechanism of attention size 512, which takes in the sequence of frame-wise feature vectors F and determines by the hidden state $\mathbf{u}_t$ how significant each frame should contribute to generate a word ($f_\theta^{(l)}(\mathbf{u}_t, F)$). Same for the region attention. So far, the architecture and hyperparameters are the same as GVD. For the rest, we used the architecture and hyperparameters selected on the F30k validation set (which is an image dataset).

Training. We train our model with the same hyperparameters as F30k. We only adjusted the learning rate for a larger batch size and reduced the number of epochs (for faster training). Namely, we set a batch size of 60 videos, a learning rate of $3e-4$ and trained for 30 epochs, reporting validation results with the model obtained at the end of training. Following [17], we uniformly sample 10 frames from each video segment during training and testing. Ground-truth captions are truncated to 20 words during training and testing.

Evaluation. We evaluate our model (the checkpoint at the end of training) on weakly-supervised object grounding and grounded captioning on the validation set. Unfortunately, the official CodaLab evaluation server[3] was closed at the time of submission. We use the official metrics and evaluation scripts for evaluating: captioning[4] and grounding[5].

B.3. Additional experimental details for YouCook2

Implementation and experimental setup for the results reported in Table 6 of the main paper:

Inputs. We use the same region proposals and features as Shi et al. [12]. For each frame, we use a Faster R-CNN [9] detector with VGG-Net [13] backbone pretrained on Visual Genome [5] to obtain region proposals. In particular, we retain the top 20 region proposals, based on their detection confidence score. Each region is described by a 4096-dimensional feature vector extracted from the $fc7$ layer of the VGG-Net. We also combine that region feature with a 300-dim trainable embedding of the bounding box coordinates (including the normalized frame index). We also use a

[3] https://competitions.codalab.org/competitions/20537

[4] https://github.com/LuoweiZhou/densevid_eval_spice/blob/bbab10c202e956266031a0dd6c791cba25b58e59/evaluate.py

[5] https://github.com/facebookresearch/ActivityNet-Entities/blob/aa5cd28383e5e9c63e875ada54057591a71509d9/scripts/eval_grd_anet_entities.py

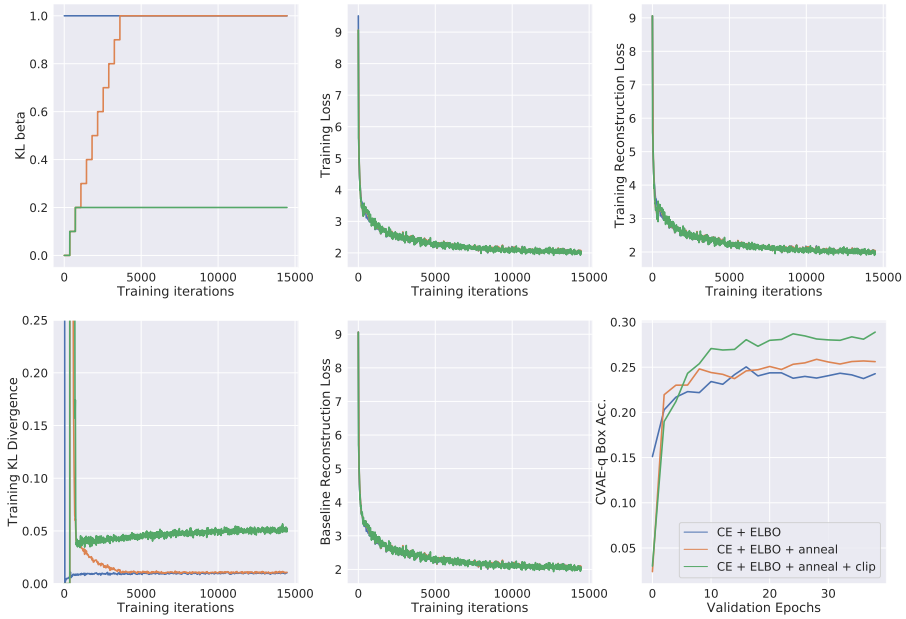


Figure 1. Comparison of learning curves for three schedules of the β hyperparameter on the Flickr30k training/validation sets. This figure supplements ablation results in Table 3 of the main paper. The clipped linear annealing schedule (green) results in higher KL divergence (the approximate posterior does not collapse to the prior) and in higher grounding accuracy. For this ablation study, we use a GVD-CVAE with a simple LSTM decoder (and no attention mechanisms k_θ and f_θ over region and grid features). We also used a simple LSTM for the inference model and $\gamma = 0$ (namely the approximate posterior sees the sentence up to the current word). This model corresponds to row 4 of Table 3 in the main paper.

	WSL Task		Method details			Box accuracy (%)		Query accuracy (%)	
	Caption.	MIL	Obj. Int.	Frm. Sim.	Reg. Sim.	macro	micro	macro	micro
Upper Bound						62.41	-	-	-
DVSA-frm [2, 12]		✓				37.55	44.16	39.31	46.14
Zhou [12, 18]		✓	✓	✓		35.08	42.42	36.69	44.34
NAFAE [12]		✓		✓	✓	40.71	46.33	42.45	48.41
STVG [16]		✓	✓	✓		41.63	47.02	43.40	48.98
SCL [14]		✓	✓			**42.80**	**48.60**	**44.61**	**50.61**
GroundR [10, 12]	✓					19.94	-	-	-
GVD-CVAE (Ours)	✓					38.00 ± 0.20	44.61 ± 0.09	40.57 ± 0.17	45.63 ± 1.80

Table 3. Grounding performance comparison on YouCook2 test set following the experimental setup of Shi et al. [12]. We compare with methods that exploit various tasks for weakly-supervised learning (WSL): captioning (Caption.) or multiple instance learning (MIL). Our captioning-based method is competitive with advanced MIL-based methods for weakly-supervised video object grounding and can additionally perform grounded captioning. Obj. Int.: modeling inter-object spatio-temporal interactions, e.g. using self-attention. Frm. Sim.: modeling word-to-frame similarity to better handle frames where the groundable word is occluded. Reg. Sim.: modeling similarity among grounded regions across frames for a groundable word. (This table is the same as Table 6 in the main paper, but with additional metrics and details about methods.)

global feature vector of size 3072 describing the video segment to be captioned, which is obtained by averaging the temporal sequence of frame-wise appearance and motion features from [19]. Following GVD [17] the global feature vector is augmented with a 50-dimensional embedding of the segment positional information (i.e., total number of segments, segment index, start time and end time). We note here that MIL-based methods in this dataset do not use that global feature vector. We use a vocabulary of 1009 words including `UNK` (the symbol for rare words not included in the vocabulary) and `EOS` (end of sentence special symbol). Words are embedded to a 512-dimensional vector using randomly initialized embeddings, trained from scratch (in contrast to Shi et al. [12], who use pre-trained GloVE word

	Box accuracy (%)		Query accuracy (%)	
	macro	micro	macro	micro
Upper Bound	62.42	68.56	65.55	70.32
GroundR [10, 12]	19.63	-	-	-
DVSA-frm [2, 12]	36.90	44.26	38.48	46.27
DVSA-vid [2, 12]	36.67	43.62	38.20	45.60
MCOG [12, 18]	35.69	43.04	37.26	44.99
NAFAE [12]	39.54	46.41	41.29	48.52
STVG [16]	39.90	46.80	41.36	48.74
SCL [14]	**41.94**	**48.46**	**43.46**	**50.45**
GVD-CVAE (Ours)	38.85	45.91	40.54	48.01

Table 4. Grounding performance comparison on YouCook2 validation set following the experimental setup of Shi et al. [12].

embeddings for the groundable words).

Model. The pre-extracted region and global video features are transformed into $\mathbf{x}$ and $\mathbf{v}$ by our trainable encoder, i.e. a pair of two linear transformation layers that project features to a lower-dimensional space of dimensionality 1024. Our decoder has at its core a single layer LSTM of hidden size 1024. Our p-attention network is an 'concat' attention mechanism with attention size 512. Our inference model consists of an BiLSTM with hidden size 1024 and a q-attention network with an 'concat' attention mechanism of size 512.

Training. We train our model for 40 epochs with the Adam [3] optimizer, having an initial learning rate of $1e-4$, decayed by a factor 0.8 when every 3 epochs. Our batch size is 80 video segments and $S = 10$, $\tau = 0.8$, and $\lambda = 0.5$. The latter were chosen based on Box accuracy on the validation set. For annealing β, we use the PI-Controller [11], with $Ki = -0.0001$, $exp_{kl} = 0.1$, $Kp = 0.01$. We apply dropout 0.5 on fully-connected layers. All layers are trained from scratch, except for the backbones yielding the initial region and video features. Following [12, 18], we randomly sample 5 frames from each video segment during training, while we use all frames (extracted at 1fps) during testing. Ground-truth captions are truncated to 20 words during training and whole captions are used during testing (maximum sentence length 46 words).

Evaluation. We evaluate our model (the checkpoint at the end of training) on the validation and testing sets using the same experimental setup and metrics as in NAFAE [12][6]. We use the CVAE prior distribution to ground each groundable word in each frame. We made this choice, since CVAE-p outperformed the CVAE approximate posterior (CVAE-q) in the validation set of this dataset. As we discussed in our main paper, our model assumes a single region grounding each word. However, in YouCook2 the model is evaluated for grounding words in every frame. This could be the reason that, in contrast to the Flickr30k Entities and ActivityNet Entities datasets, the CVAE-q (with macro box accuracy 35.8%) does not clearly outperform CVAE-p (with macro box accuracy 38.85%) in this dataset that evaluates grounding in each frame.

[6] https://github.com/jshi31/NAFAE/blob/master/lib/datasets/youcook_eval.py

B.4. Software

All models were implemented in Python using Pytorch and are based on the Grounded Video Description (https://github.com/facebookresearch/grounded-video-description) open-source code. Given pre-extracted video and region features, a forward pass through our model for performing grounding on 20 ActivityNet videos (10 frames sampled from each, $M = 1000$) takes 0.7 seconds at a single Tesla K80 GPU. We train our models on 4 GPUs and training lasts from around 6 to 24 hours depending on the dataset (training on the ActivityNet Entities video datasets lasts longer than trianing on the Flickr30k Entities image dataset). These are the websites for the public, benchmark datasets that we used in this work.

- Flickr30k Entities http://bryanplummer.com/Flickr30kEntities/
- ActivityNet Entities https://github.com/facebookresearch/ActivityNet-Entities
- YouCook2-BB http://youcook2.eecs.umich.edu/download

C. Additional qualitative results

In this section we discuss in Fig. 2 some of the qualitative results presented in Figure 4 of the main paper. We would like to emphasize that these qualitative results used a simpler model. It uses a single LSTM in the decoder, single LSTM in the inference model and $\gamma = 0$. It also anneals β using a clipped linear schedule instead of the PI-Controller. (It is the same model in row 4 of Table 3 in the main paper). Qualitative results on ActivityNet Entities are demonstrated in Fig. 3.

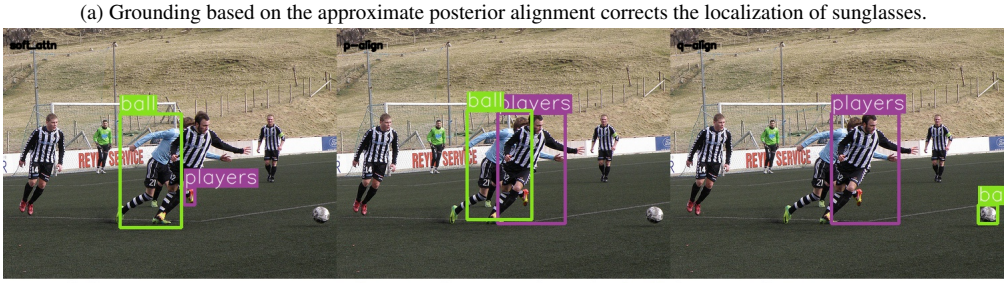
References

[1] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. 1

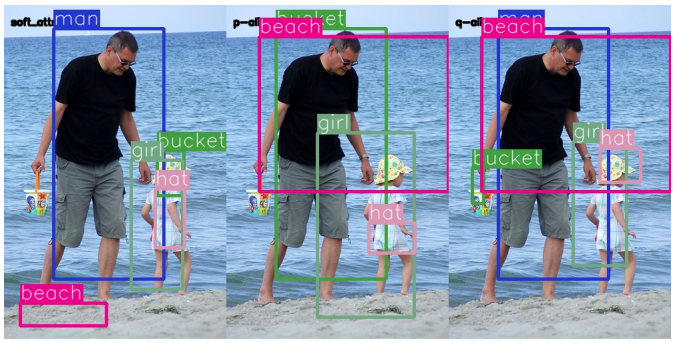
[2] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:664–676, 4 2017. 5, 6

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3, 6

(a) Grounding based on the approximate posterior alignment corrects the localization of sunglasses.



(b) Our p-attention network (parameterizing the prior alignment) can correctly ground players, while additionally conditioning on the groundable words corrects the localization of the ball.



(c) Our GVD-CVAE can accurately localize the small objects: bucket and hat.

(d) Failure case: Our approximate posterior alignment fails to disambiguate between the two men and the two shirts. In contrast, our prior alignment which grounds based on: "[...] in a yellow", accurately localizes the shirt. Grounding based on whole phrases (yellow shirt) instead of individual words might help mitigate this issue.

Figure 2. Qualitative comparison of weakly-supervised object grounding results obtained by the baseline and our GVD-CVAE on images from Flickr30k Entities. For each caption, we show three copies of each image with grounding results obtained by the soft-attention baseline, our prior and posterior alignment distributions, respectively. For this result, we use a GVD-CVAE with a simple LSTM decoder (and no attention mechanisms k_θ and f_θ over region and grid features). We also used a simple LSTM for the inference model and $\gamma = 0$ (namely the approximate posterior sees the sentence up to the current word). This model corresponds to row 4 of Table 3 in the main paper.

[4] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. In *Foundations and Trends in Machine Learning*, 2019. 1

[5] 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. 2, 3, 4

[6] Miao Liu, Siyu Tang, Yin Li, and James M Rehg. Forecasting human-object interaction: joint prediction of motor attention and actions in first person video. In *European Conference on Computer Vision*, pages 704–721. Springer, 2020.

[7] Chih-Yao Ma, Yannis Kalantidis, Ghassan AlRegib, Peter Vajda, Marcus Rohrbach, and Zsolt Kira. Learning to generate grounded visual captions without localization supervision. In *European Conference on Computer Vision*, 2020. 4

[8] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017. 1

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. 2, 3, 4

[10] Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision*, 2016. 5, 6

[11] Huajie Shao, Shuochao Yao, Dachun Sun, Aston Zhang, Shengzhong Liu, Dongxin Liu, Jun Wang, and Tarek Abdelzaher. Controlvae: Controllable variational autoencoder. In *International Conference on Machine Learning*, 2020. 3, 6

[12] Jing Shi, Jia Xu, Boqing Gong, and Chenliang Xu. Not all frames are equal: Weakly-supervised video grounding with contextual similarity and visual clustering losses. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 4, 5, 6

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations*, 2015. 4

[14] Wei Wang, Junyu Gao, and Changsheng Xu. Weakly-supervised video object grounding via stable context learning. In *ACM International Conference on Multimedia*, New York, NY, USA, 2021. Association for Computing Machinery. 5, 6

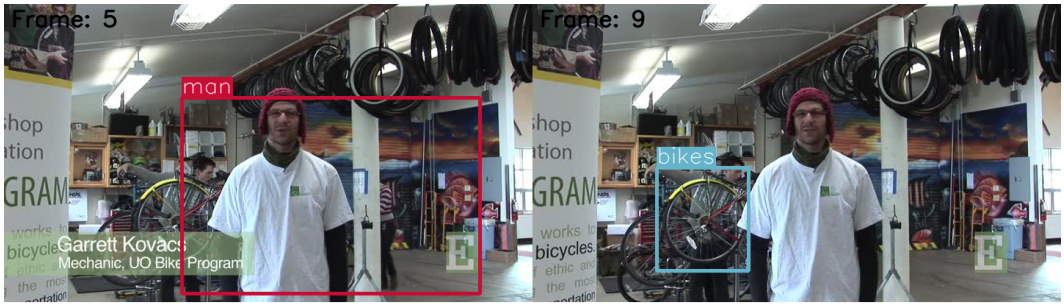
[15] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, July 2017. 2, 3

[16] Xun Yang, Xueliang liu, Meng Jian, Xinjian Gao, and Meng Wang. Weakly-supervised video object grounding by exploring spatio-temporal contexts. In *ACM International Conference on Multimedia*, New York, NY, USA, 2020. Association for Computing Machinery. 5, 6

[17] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J. Corso, and Marcus Rohrbach. Grounded video description. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2019-June, pages 6571–6580. IEEE Computer Society, 6 2019. 1, 2, 3, 4, 5

[18] Luowei Zhou, Nathan Louis, and Jason J. Corso. Weakly-supervised video object grounding from text by loss weighting and object interaction. In *British Machine Vision Conference*, 2019. 1, 5, 6

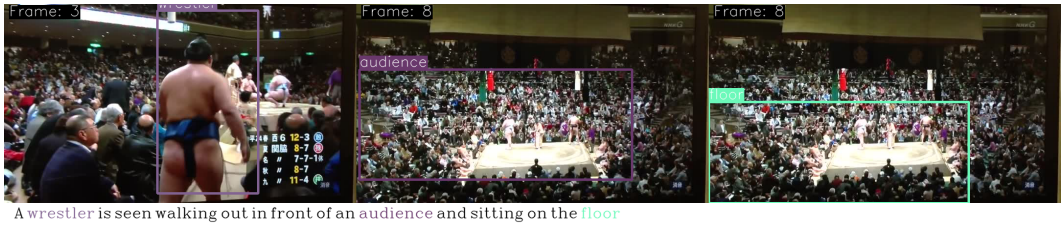
[19] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 5



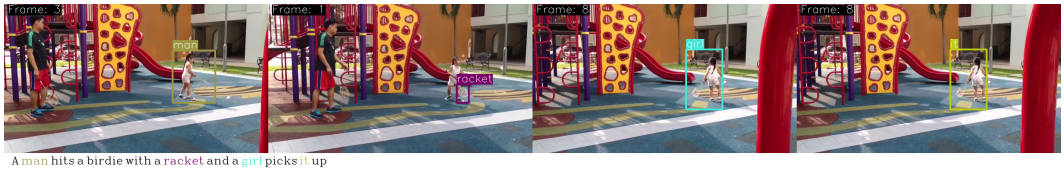
(a) Failure case: Although the man is localized, the bounding box is not tight enough. This is common because of the lack of bounding box annotations. Moreover, singular and plural forms of words are converted to the same representation during training and testing, leading to sub-optimal grounding of groups of objects.

(b) Kids and dogs are accurately localized.

(c) The woman and the accordion are correctly localized.



(d) Note how the model seems to ground the words at the frames that they are visible.



(e) Failure case: the model fails to ground the correct racket and the man. Modeling the dependencies between the regions grounding each word in the sentence might help mitigate such issues.

Figure 3. Qualitative weakly-supervised object grounding results obtained on videos from the ActivityNet Entities validation set. For each groundable word in a ground-truth caption, we show the aligned region that is the mode of the approximate posterior distribution (region with the maximum q-attention network coefficient over all regions in the 10 equally-spaced frames). For this qualitative result, we use a GVD-CVAE with a simple LSTM decoder (and no attention mechanisms k_θ and f_θ over region and grid features). We also used a simple LSTM for the inference model and $\gamma = 0$ (namely the approximate posterior sees the sentence up to the current word). The grounding accuracy of this model on ground-truth sentences of the validation set is 18% (our full model achieves a grounding accuracy of 24.2%, as reported in Table 5 of the main paper).