ROUTING WITH SELF-ATTENTION FOR MULTIMODAL CAPSULE NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

The task of multimodal learning has seen a growing interest recently as it allows for training neural architectures based on different modalities such as vision, text, and audio. One challenge in training such models is that they need to jointly learn semantic concepts and their relationships across different input representations. Capsule networks have been shown to perform well in context of capturing the relation between low-level input features and higher-level concepts. However, capsules have so far mainly been used only in small-scale fully supervised settings due to the resource demand of conventional routing algorithms. We present a new multimodal capsule network that allows us to leverage the strength of capsules in the context of a multimodal learning framework on large amounts of video data. To adapt the capsules to large-scale input data, we propose a novel routing by self-attention mechanism that selects relevant capsules which are then used to generate a final joint multimodal feature representation. This allows not only for robust training with noisy video data, but also to scale up the size of the capsule network compared to traditional routing methods while still being computationally efficient. We evaluate the proposed architecture by pretraining it on a large-scale multimodal video dataset and applying it on four datasets in two challenging downstream tasks. Results show that the proposed multimodal capsule network is not only able to improve results compared to other routing techniques, but also achieves competitive performance on the task of multimodal learning.

1 INTRODUCTION

With the proliferation of video sharing websites and affordable recording devices, the amount of video data available today has dramatically increased. Given that hand annotating this continuously growing stream of data is infeasible, recent research has turned to training networks on such large-scale multimodal data without manual annotation (Miech et al., 2019; Alwassel et al., 2019; Miech et al., 2020). These works make use of the fact that large amounts of data are available across multiple modalities such as vision, text, and audio, especially like in case of videos. This data has been used in two ways: first to learn feature representations from video data by pretraining on large datasets (Alwassel et al., 2019), and second to train networks that are able to relate cross-modal inputs based on the similarity of their internal neural representations (Miech et al., 2020; Alayrac et al., 2020), which can be applied to zero-shot tasks like classification or text-to-video retrieval. Especially for the latter case, it becomes necessary to capture similar semantic relationships across very different and low level feature representations, as e.g. video features extracted by a ResNet architecture (He et al., 2016) have to be related to bag of words representations of sentences (Miech et al., 2020), or even sound representations extracted from audio waveforms (Rouditchenko et al., 2021). These relationships can be captured by training a network that takes pairs of modalities as inputs and predicts a similarity score, or by projecting both representations into a joint embedding space. In the second case, for example, the encoding for a sentence like "Cut the chicken." would be close to the encoding of the visual representation of frames showing this activity and further away from the encoding of frames showing other objects like vegetables or unrelated topics like outdoor activities. The semantic closeness can then be measured based on distance metrics.

Learning such a joint embedding space involves the grouping of similar concepts across different modalities. Here, it can be helpful to identify which low-level features show activation in certain contexts, which can serve as a form of filtering to focus on relevant inputs and thus learn a good joint

embedding space. Capsule networks (Sabour et al., 2017) have been proposed as a technique to capture activations of a specific type of entity and to model higher-level objectness from groups of low-level feature activations. To this end, capsule networks find familiar concepts by performing "high-dimensional coincidence filtering" (Hinton et al., 2018) through a routing-by-agreement algorithm. They have shown their ability in modeling these relationships in images (Sabour et al., 2017; Hinton et al., 2018; Kosiorek et al., 2019) and videos (Duarte et al., 2018), and have also performed well in multimodal applications (Urooj Khan et al., 2021; McIntosh et al., 2020). However, these applications have mainly been used for learning in a fully supervised setting with clean data.

In this work, we leverage the qualities of capsule architectures in the context of multimodal learning to learn a joint embedding space across different input modalities. To allow the capsules to learn from large-scale noisy input data, we propose an efficient routing by self-attention mechanism that finds similarity between these lower-level capsule representations to produce higher-level capsules and activations. To this end, we build upon the standard capsule network setup and generate a set of capsules for the input features. From these capsules, we obtain votes for higher-level capsules, in the form of key-query-value tuples, and perform a self attention operation to obtain the higher-level capsule pose representations. These are then passed through a linear layer and a softmax layer to obtain the final activations. These activations are used to select relevant capsules, increasing the impact of those feature groups belonging to certain object representations while reducing the impact for irrelevant ones. We find that this self-attention based routing mechanism is more scalable than standard dynamic (Sabour et al., 2017) and EM Hinton et al. (2018) routing methods: this is vital for applying capsule networks to large-scale video datasets.

The proposed multimodal capsule network is trained by mapping the selected capsules to a joint multimodal embedding space which is enforced by the use of a contrastive loss. For evaluation, we train the system on the HowTo100M multimodal dataset, consisting of 1.2 million YouTube instruction videos and evaluate the resulting method on the two zero-shot down-stream tasks of video retrieval on the YouCook2 (Zhou et al., 2018) and MSR-VTT (Xu et al., 2016) dataset and action localization on the CrossTask (Zhukov et al., 2019) and the MiningYouTube (Kuehne et al., 2019) dataset. Our experiments show that the proposed architecture is able to improve performance compared to existing routing mechanisms and to provide competitive performance on all evaluated downstream tasks.

The contributions of the paper are as follows:

- We propose a novel routing by self-attention mechanism for capsule architectures.
- We show that the proposed mechanism is more efficient and scalable compared to other routing techniques.
- To the best of our knowledge, we are the first to evaluate different capsule architectures on large-scale multimodal data without human annotation.

2 Related Work

Capsule Networks The concept of capsule networks was first introduced in (Hinton et al., 2011), where view-equivariant vector representations were learned from images. Sabour *et al.* (Sabour et al., 2017) extended this idea and proposed an iterative routing-by-agreement algorithm which was able to classify and segment overlapping digits. Capsule network have been widely applied to various domains and problems. including text classification (Yang et al., 2018), video action detection (Duarte et al., 2018), point cloud processing (Zhao et al., 2019), and medical image segmentation (LaLonde et al., 2021). One key aspect of capsules networks is their ability to route, and in a hierarchical fashion activate higher-level capsules based on agreement of multiple lower-level capsules. However, this ability comes with the increased computational cost of the routing-by-agreement algorithm. First capsule architectures (Sabour et al., 2017) used dynamic routing which can be computationally slow and results in high memory consumption, especially on higher dimensional input data and for a larger number of capsules. Hinton et al. (2018) reduced the number of parameters in their capsule network by learning matrix capsules with an iterative Expectation-Maximization (EM) based routing algorithm. Several other works have attempted to make more efficient and scalable routing algorithms including self-routing capsule networks (Hahn et al., 2019), KDE-based routing (Zhang et al., 2018),

STAR-Caps (Ahmed & Torresani, 2019), spectral capsule networks (Bahadori, 2018), and subspace capsule networks (Edraki et al., 2020).

Recently, Efficient-CapsNet (Mazzia et al., 2021) and stacked capsule autoencoders (SCAEs) (Kosiorek et al., 2019) have proposed routing mechanisms based on attention. Efficient-CapsNet uses vector capsules similar to those found in Sabour et al. (2017), and computes self-attention across votes of the lower-level capsule layers to find the routing/coupling coefficients. Then, the resulting higher-level capsules are a weighted sum across these same votes based on the coefficients. On the other hand, our proposed self-attention routing method generates separate key, query, and value representations. This allows us to separate the computation of the votes and the routing coefficients: the key and query generate the routing coefficients, which are used to weight the votes to obtain the higher-level capsules. SCAEs adapt the Set Transformer (Lee et al., 2019) to perform routing between the set of part capsules to the object capsules. They generate capsule activations by maximising the part pose likelihood from a mixture of predictions from lower-level capsules. To adapt the concept of attention-based routing to the problem of multimodal learning from large-scale noisy data, we propose to use self-attention instead of Set Transformer and generate the query by linear projection from the input capsules. Additionally, different from stacked capsule autoencoders, we solely rely on the self-attention mechanism for routing and compute activation by the linear transformation of the higher level features followed by a softmax. We found this setting is computationally more efficient and allows us to train without the need for tuning of sparsity constraints, while showing higher performance compared in the targeted setup (see Sec. 4.4).

Multimodal Learning As annotating large datasets (Deng et al., 2009; Carreira & Zisserman, 2017) is extremely costly, recent approaches take advantage of the vast amount of video data on websites and social media platforms. By leveraging readily available tools like automatic speech recognition systems, narrated video datasets can be constructed (Miech et al., 2019; Sanabria et al., 2018) on which proxy tasks can be used to learn meaningful representations. Different methods have been proposed to learn from these video/text pairs (Amrani et al., 2020; Gabeur et al., 2020; Luo et al., 2020; Zhu & Yang, 2020; Patrick et al., 2020; Lei et al., 2021; Sun et al., 2019; Dong et al., 2019) as well as from video/audio pairs (Alwassel et al., 2020; Asano et al., 2019; Boggust et al., 2019; Rouditchenko et al., 2021), and from all three modalities (video, audio, and text) (Alayrac et al., 2020). These self-superivsed multimodal approaches tend to rely on convolutional or transformer architectures; to the best of our knowledge, this is the first work which employs capsule networks in this problem. Most methods use the large-scale data for pretraining the network followed by a fine-tuning on a downstream dataset, which is usually done with less noisy curated or hand-annotated data (Luo et al., 2020; Patrick et al., 2020; Lei et al., 2021; Alwassel et al., 2020; Rouditchenko et al., 2021; Dong et al., 2019). However, some approaches show that the training on large-scale noisy data alone can also be sufficient and directly apply the resulting model without fine-tuning to the downstream datasets (Amrani et al., 2020; Gabeur et al., 2020; Patrick et al., 2020; Sun et al., 2019; Boggust et al., 2019).

3 Multimodal Learning with Capsule Networks

In the following, we first describe the proposed multimodal capsule architecture (Figure 1) at high level, and then follow with a detailed description of the proposed routing by self-attention mechanism depicted in Figure 2. We close with a description of the training procedure.

3.1 System setup

Given n video clips, each with a corresponding video, audio, and text representations we attempt to learn a joint multimodal representation space. We denote the video as $v \in \mathcal{V}$, the audio as $a \in \mathcal{A}$, and the text narration generated by an automated speech recognition (ASR) system as $t \in \mathcal{T}$. Thus, the training set of n video clips is representated by tuples $\{(v_i, a_i, t_i)\}_{i=1}^{n}$. The goal of contrastive multimodal learning is to learn a set of functions to generate embeddings for each modality such that embeddings for semantically similar inputs are closer together than semantically dissimilar inputs. Formally, we learn functions, $f_v : \mathcal{V} \rightarrow \mathbb{R}^D$, $f_a : \mathcal{A} \rightarrow \mathbb{R}^D$, and $f_t : \mathcal{T} \rightarrow \mathbb{R}^D$ which create D dimensional embeddings (i.e. $f_v(v) \in \mathbb{R}^D$). The input representations take the form of pre-extracted 2D and 3D features from a video clip, log-mel spectrograms extracted from an audio segment, and a

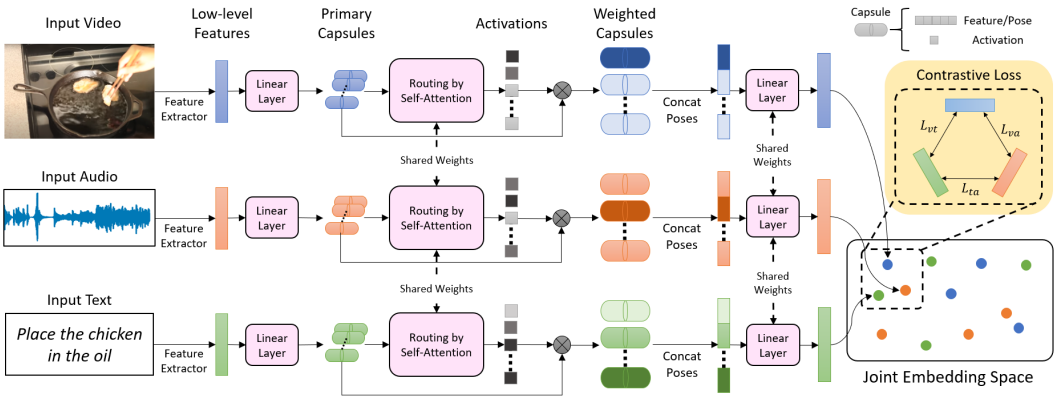

Figure 1: **Overview of our proposed approach.** Given a video, audio, and text triplet, the network extracts modality specific features and converts them into a set of primary capsules. Then, these capsules are routed using self-attention to obtain a higher-level activations, which are used to weight capsule features. The weighted capsule features are projected into a final joint multimodal feature representation. This joint representation space is enforced by a pair-wise contrastive loss.

text embedding extracted by sentence-based neural network. The goal is to find mapping functions f_v, f_a, and f_t, so that the distance of all possible pairs from the same tuple (v_i, a_i), (t_i, a_i), and (v_i, t_i) is minimized in the embedding space and the distance to all other tuple pairs is maximized. An overview of the overall system is shown in Figure 1.

3.2 Multimodal Capsule Architecture

Primary Capsules To learn the mapping of each input feature to the joint embedding space, i.e. functions f_v, f_a, and f_t, we propose a novel capsule network architecture. From each input modality feature, a learned linear layer extracts a set of C primary capsules. A capsule is composed of a d-dimensional pose vector x, which represents an entity's properties and an activation p, which represents an existence probability (i.e. the probability that the given entity/object exists within the input). The i-th capsule for modality m has the pose vector $x_i^m \in \mathbb{R}^{d_1}$ and activation $p_i^m \in [0, 1]$. We use these capsules in a self-attention based routing-by-agreement algorithm, depicted in Figure 2, to learn the relationships between the different entities they model.

Routing by Self-Attention We first multiply capsule pose vectors x_i^m by their respective activations p_i to ensure entities which are not present (i.e. $p_i^m \to 0$) are not used in the routing process. We then learn a set of functions to extract the respective key, query, and value representations from capsules $K = h_K(p_i^m x_i^m), Q = h_Q(p_i^m x_i^m), V = h_V(p_i^m x_i^m)$, using two linear layers for all functions h. These learned functions, $h_K, h_Q, h_V : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$, map the primary capsules pose vectors to the secondary capsules' pose feature space and are used in a multi-head self-attention mechanism:

$$\hat{x}_i^m = \text{Attention}\,(Q, K, V) = g\left(\text{softmax}\left(\frac{QK^T}{\sqrt{d_2}}\right)V\right), \tag{1}$$

where g is a two-layer nonlinear MLP[1]. In the context of capsule routing, the query and key produce the routing coefficients which determine the amount of information a lower-level capsule sends a specific higher level capsule, whereas the value can be considered a vote, or prediction, for the properties of the higher level capsule.

From the secondary capsule layer's poses, $\hat{x}_i^m$, we generate their existence probabilities, through a softmax operation:

$$\hat{p}_i^m = \frac{\exp\left(x_i^m W_p + b_p\right)}{\sum_{j=1}^{C} \exp\left(x_i^m W_p + b_p\right)}, \tag{2}$$

where $W_p \in \mathbb{R}^{d_2 \times 1}$ and $b_p \in \mathbb{R}$ are learned parameters.

[1]See Appendix B for additional details

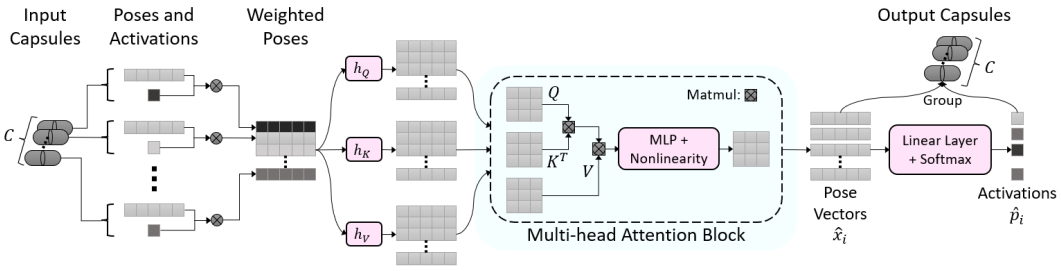

Figure 2: **Proposed Routing by Self-Attention.** The input is a set of C capsules. The activation-weighted capsule features are projected into query, key, and value matrices which are used in a multi-head self-attention block to generate higher-level capsule poses. A linear transformation with softmax activation then generates the activations for these higher-level capsules.

Mapping to Joint Embedding Space These existence probabilities are used to select relevant capsules. This is done by passing the activation-weighted capsules through a linear transformation, f_{out} to obtain the final feature representations that are used in the final loss:

$$f_v = f_{\text{out}}\left(\hat{p}_i^v x_i^v\right), f_a = f_{\text{out}}\left(\hat{p}_i^a x_i^a\right), \text{ and } f_t = f_{\text{out}}\left(\hat{p}_i^t x_i^t\right). \tag{3}$$

Note that all learned weights used after the generation of the primary capsule layer, namely h_K, h_Q, h_V and f_{out}, are shared across modalities.

3.3 Contrastive Multimodal Learning

To train the described architecture and learn the joint representation space, we use a contrastive loss on each pair of modalities (v, a), (t, a), and (v, t). For different modalities from the same video clip, the contrastive loss maximizes the similarity of their embeddings; conversely, it minimizes the similarity for embeddings from different video clips. Following Rouditchenko et al. (2021), we use the Masked Margin Softmax (MMS) loss (Ilharco et al., 2019), which defines the dot-product between two vectors as the similarity measure and computes similarities across a batch of B samples. The loss is computed between two modalities, and can be viewed as the sum of two instances of InfoNCE (Oord et al., 2018) (with a margin δ). For example, the loss for the visual/audio pair (L_{va}) consists of two components: the first where the visual input is fixed and audio samples are varied, and the second where the audio input is fixed and visual samples are varied. We sample negatives from both within the same video and from other videos, since this has been shown to empirically improve performance (Miech et al., 2019). The final loss is the sum of the pairwise MMS losses between different modalities:

$$L_{\text{final}} = L_{va} + L_{ta} + L_{vt}. \tag{4}$$

Since the loss is computed over all modality pairs, it ensures all features are projected into the same space and are comparable.

4 Experimental Evaluation

In this section, we assess the performance of the proposed approach in the context of multimodal learning. For this evaluation, we focus on the zero shot capabilities of the proposed approach, namely on the downstream tasks of zero-shot text-to-video retrieval and zero-shot temporal action localization, as this allows us to evaluate how well high-level semantic concepts have been identified and grouped across various modalities. We first present an overview on the implementation details of our proposed approach. The overall system performance is then compared with various other techniques in the field. We finally evaluate the impact of each component including the routing mechanism in comparison with other available techniques and present qualitative results for the proposed method. The code and related resources will be made publicly available to allow for reproducability of presented results.

4.1 Implementation Details

Following Miech et al. (2019), the input visual features for our method are 2D features extracted at 1 fps using a ResNet-152 model (He et al., 2016) pretrained on ImageNet (Deng et al., 2009), as well

Table 1: Evaluation of zero-shot text-to-video retrieval. MIL-NCE* uses the same training procedure as (Miech et al., 2020) with different backbone features, † indicates trainable backbone. Modality indicates the modalities used during inference, where V: video, T: text, A: audio.

			YouCook2				MSR-VTT			
Method	Modality	Visual Backbone	R@1↑	R@5↑	R@10↑	MedR↓	R@1↑	R@5↑	R@10↑	MedR↓
MMT (Gabeur et al., 2020)	VT	7 experts	-	-	-	-	-	14.4	-	66
ActBERT (Zhu & Yang, 2020)	VT	R101+Res3D	9.6	26.7	38.0	19	8.6	23.4	33.1	36
Support Set (Patrick et al., 2020)	VT	R152+R(2+1)D-34	-	-	-	-	8.7	23.0	31.1	31
MIL-NCE (Miech et al., 2020)†	VT	I3D-G	11.4	30.6	42.0	16	9.4	22.0	30.0	35
MMV FAC (Alayrac et al., 2020)†	VAT	TSM-50x2	11.7	33.4	45.4	13	9.3	23.0	31.1	38
HT100M (Miech et al., 2019)	VT	R152+RX101	6.1	17.3	24.8	46	7.2	19.2	28.0	38
NoiseEstimation (Amrani et al., 2020)	VT	R152+RX101	-	-	-	-	8.0	21.3	29.3	33
MIL-NCE* (Miech et al., 2020)	VT	R152+RX101	8.0	22.9	32.1	**29**	8.6	23.1	**30.8**	33
Ours	VT	R152+RX101	**9.0**	**23.2**	**32.5**	30	**9.7**	**23.2**	30.7	**32**
AVLNet (Rouditchenko et al., 2021)	VAT	R152+RX101	**19.9**	36.1	44.3	16	8.3	19.2	27.4	47
Ours	VAT	R152+RX101	19.3	**37.8**	**47.3**	**13**	**9.3**	**21.4**	**30.9**	**37**

as 3D features extracted at 1.5 fps using a ResNext-101 (Hara et al., 2018) pretrained on Kinetics (Carreira & Zisserman, 2017). These features are max-pooled over time and concatenated to form a 4096 dimension feature vector for a given video clip. The audio input to our network are features extracted from the log-mel spectrograms by a pre-trained DAVEnet model (Harwath et al., 2018). For textual features, we follow Miech et al. (2019), and use a GoogleNews pretrained Word2vec model (Mikolov et al., 2013) to extract word embeddings. Then a max-pooling operation over all word embeddings in a sentence produces a single vector representation. All feature extraction backbones are fixed (i.e. not fine-tuned) during training and evaluation. To train our network we use an Adam optimizer (Kingma & Ba, 2014) with a learning rate of 0.001 and cosine learning rate scheduler (Misra & Maaten, 2020). The model is trained on 4 V100 GPUs for 20 epochs, using a batch size of 4096. In the MMS loss, we set $\delta = 0.001$. Unless otherwise stated, we set the number of capsules to $C = 128$, the dimension of each capsule's pose vector to $d_1 = 32$ and $d_2 = 256$, and the final joint representation dimension is $D = 4096$. Our method is trained using the HowTo100M (Miech et al., 2019) instructional video dataset, which consists of 1.2 million videos with corresponding audio and text transcripts extracted using an off-the-shelf ASR system. The video-audio-text tuples are defined by the transcription timestamps provided with the dataset.

4.2 Text-To-Video Retrieval

Datasets and Metrics The problem of text-to-video retrieval involves searching a pool of videos for a single video that corresponds to a given ground-truth text query. We evaluate zero-shot text-to-video retrieval on the YouCook2 (Zhou et al., 2018) and MSR-VTT (Xu et al., 2016) datasets, which are common benchmark datasets for zero-shot video retrieval. The YouCook2 dataset consists of cooking instructional video clips with human-annotated text descriptions, and we use the validation set of 3.5k clips following prior work (Miech et al., 2019; 2020). The MSR-VTT dataset contains 10K video clips with human-annotated captions on various topics, and we use the test set of 1K video clips from (Miech et al., 2019). For the retrieval task, we compute the euclidean distance between the text and video representations through the pretrained network to find e.g. the top video candidates for a given text sample. For both datasets, we evaluate using the recall metrics: R@1, R@5, R@10, and Median Recall (MedR).

Comparison with the state-of-the-art We report the results on the text-to-video retrieval task for YouCook2 and MSR-VTT in Table 1 for two cases, zero-shot text-to-video retrieval (VT) and zero-shot text-to-video+audio retrieval (VAT). We find that our method outperforms prior approaches which use the same video backbone in both cases and on both downstream datasets. Interestingly, the addition of the audio modality leads to a large performance boost on YouCook2, but seems to decrease performance on MSR-VTT. This can be attributed to the domain shift between the ASR generated text and its correlation to audio in HowTo100M, and the less correlated audio and text in MSR-VTT, where captions are hand-generated without instructional focus or alignment to the sound. On the other hand, the audio and text present in YouCook2 more closely resemble the training data, leading to improved performance.

Table 2: Evaluation of zero-shot temporal action localization. MIL-NCE* uses the same training procedure as (Miech et al., 2020) with different backbone features, † indicates trainable backbone. Modality indicates the modalities used during inference, where V: video, T: text, A: audio.

		CrossTask			MYT		
Method	Visual Backbone	Recall↑	IOD↑	IOU↑	Recall↑	IOD↑	IOU↑
Cross-task (superv.) (Zhukov et al., 2019)	R152+I3D	31.6	-	-	-	-	-
Cross-task (weakly superv.) (Zhukov et al., 2019)	R152+I3D	22.4	-	-	-	-	-
ActBERT (Zhu & Yang, 2020)	R101+Res3D	37.1	-	-	-	-	-
ActBERT (Zhu & Yang, 2020)	+ Faster R-CNN	41.4	-	-	-	-	-
MIL-NCE (Miech et al., 2020)†	I3D-G	36.4	-	-	-	-	-
Mining: MLP (weakly superv.) (Kuehne et al., 2019)	TSN	-	-	-	-	19.2	9.8
HT100M (Miech et al., 2019)	R152+RX101	33.6	26.6	17.5	15.0	17.2	11.4
MIL-NCE* (Miech et al., 2020)	R152+RX101	33.2	30.2	16.3	14.9	26.4	17.8
Ours	R152+RX101	**35.2**	**32.6**	**21.4**	**18.0**	**31.6**	**22.9**

Table 3: Evaluation of different types of routing functions as well as without routing for $C = 64$ number of capsules and a dimensionality of $d_1 = d_2 = 16$ including runtime and memory usage.

	YouCook2		MSRVTT			
Method	R@1	R@10	R@1	R@10	Memory Usage (GB)	Run-time (sec/batch)
No Routing	15.3	41.9	7.6	30.1	9.12	0.687
Dynamic Routing (Sabour et al., 2017)	17.0	**44.3**	8.2	31.1	20.50	1.534
EM Routing (Hinton et al., 2018)	5.8	24.2	5.7	21.8	19.13	1.272
Set Transformer (Lee et al., 2019)	16.5	40.0	8.4	30.0	9.11	0.707
Self-Attention (ours)	**18.6**	44.0	**8.7**	**31.6**	9.11	0.722

4.3 Temporal Action Localization

Datasets and Metrics Given a set of action classes, the goal of temporal action localization is to predict the actions present at each time-step of the video. In this task, we compute the distance between the video representation and each action's text representation to obtain a class prediction for each time-step of the video. We evaluate on the CrossTask (Zhukov et al., 2019) and Mining YouTube (Kuehne et al., 2019) datasets. CrossTask contains 2.7k instructional videos; each video frame is manually annotated using action steps/ordering for each task collected from *wikiHow*. The recall is calculated using the same inference procedure of (Zhukov et al., 2019). The Mining YouTube dataset contains videos from five simple cooking recipes - "eggroll", "fried egg", "pancake", "omelet", and "scrambled egg". The test set contains 50 videos from each task (250 in total) that are densely annotated with 512 classes comprised of verb-object pairs (94 unique verbs and 171 unique objects). For evaluation, we report the recall metric as well as the intersection over detection (IoD) (Bojanowski et al., 2014) and intersection over union (IoU) metrics as outlined in (Kuehne et al., 2019). The IoD metric is defined as $\frac{G \cap D}{D}$ and the IoU metric is defined as $\frac{G \cap D}{G \cup D}$, where G is the ground-truth action and D is the prediction.

Comparison with the state-of-the-art We present the results for the temporal action localization task in Table 2. When compared to methods with the R152+RX101 backbone feature extractor, (Miech et al., 2019; 2020), we show improved performance across both datasets and all metrics. On CrossTask, MIL-NCE (Miech et al., 2019) achieves improved recall with stronger backbone features and ActBERT (Zhu & Yang, 2020) uses a stronger language model as well as region-based features extracted by a Faster R-CNN. Furthermore, our method outperforms the fully supervised baseline in (Zhukov et al., 2019) and the state-of-the-art weakly supervised approach (Kuehne et al., 2019) on the reported metrics in CrossTask and Mining YouTube, respectively.

4.4 Ablations

Here, we present ablations to evaluate our proposed self-attention based routing mechanism's efficacy, its ability to scale with more capsules, and compare with other architectural baselines. Appendix A contains additional ablations.

Routing We compare the proposed self-attention routing with previous routing methods including dynamic (Sabour et al., 2017), EM (Hinton et al., 2018), and Set Transformer (Lee et al., 2019) routing,

Table 4: Evaluation on different number of capsules for a dimensionality of $d_1 = 32$ and $d_2 = 256$. It shows that on the given dataset we reach saturation around $C = 128$ capsules.

	YouCook2		MSRVTT			
Method	R@1	R@10	R@1	R@10	Memory Usage (GB)	Run-time (sec/batch)
$C = 32$	18.5	45.0	8.0	29.2	9.15	0.730
$C = 64$	18.1	46.1	8.6	29.4	11.24	0.768
$C = 128$	19.3	47.3	9.3	30.9	15.97	0.879
$C = 256$	18.7	46.5	8.7	30.5	27.98	1.096

Table 5: Evaluation using fully connected and self-attention baselines.

	YouCook2				MSR-VTT			
Modalities	R@1	R@5	R@10	Med. R	R@1	R@5	R@10	Med. R
Fully Connected	15.5	31.0	40.0	22	7.8	17.8	25.2	50
Self-Attention	13.8	29.3	36.2	34	8.9	22.3	30.1	41
Ours	19.3	37.8	47.3	13	9.3	21.4	30.9	37

as well as with a setup without any routing (i.e. learning a MLP to obtain existence probabilities). As dynamic and EM routing involve a computationally expensive iterative procedure and EM routing requires matrix capsules, we reduce the size of the network and fix the number of capsules to $C = 64$ and the dimensionality of the primary and secondary capsule to $d_1 = d_2 = 16$ to allow for a training with same batch size for all approaches. From the results shown in Table 3, we see that training with routing tends to outperform the respective baseline architectures without routing mechanisms. Among the evaluated methods, only the EM routing algorithm does not seem to be well suited for the targeted setup, as it greatly suffers from instability during training. Overall, the proposed routing by self-attention outperforms previous routing algorithms, closely followed by dynamic routing which also achieved relatively strong performance in this experimental setup. One problem with iterative routing procedures, including dynamic routing, is that it becomes difficult to scale, mainly because of the larger memory footprint. Here, especially in the direct comparison with dynamic routing, the proposed method is able to achieve better results with fewer computational resources.

Number of Capsules To show the ability of the proposed routing mechanism to scale, we also analyse how the number of capsules effects our proposed architecture. For these experiments, we maintain the capsule dimension of the original training setting with $d_1 = 32$ and $d_2 = 256$ while varying the number of capsules, $C = 32, 64, 128, 256$. As shown in Table 4, increasing the number of capsules generally leads to an improvement in performance. This can be seen as a indicator that a larger number of capsules allows the network to capture more object representations. With the current dataset, we find that our models saturate at $C \geq 128$; when the number of capsules becomes larger, we find that there is a diminishing return on performance. Considering computational efficiency, it further shows that even for large numbers of capsules, the run-time is still below the run-times of the iterative routing mechanisms on smaller sets.

Comparison with Fully Connected and Self-Attention Baselines Since our main contribution is the proposal of a capsule-based framework for multimodal learning, we compare with other architecture baselines in Table 5. For a standard baseline, we have run an experiment which takes the input features and passes them through two fully connected layers (Fully Connected). It achieves lower performance than our proposed capsule network, showing that using capsules is valuable in learning multi-modal representations. We also compare with self-attention without capsule structures. For this experiment, we apply a multi-head self-attention layer on the input features. We take the input features and group the activations into N equal length vectors. Here $N = 128$ so that it is as similar to the number of capsules in our main experiments. These vectors are used as the sequence for a self-attention layer, which is followed by a fully-connected layer to obtain the final feature representation for each modality. We find that self-attention outperforms the fully connected baseline on MSR-VTT, but does not reach the performance of our proposed self-attention routing method.

4.5 Qualitative Analysis

In our final set of evaluations, we attempt to understand what the proposed architecture is able to learn by analysing a set of qualitative retrieval examples as well as studying how individual capsule activations effect the final feature representation.

Figure 3: Qualitative retrieval examples: top-3 zero-shot text-to-video retrieval results on the YouCook2 dataset for the proposed approach with self-attention based routing, the same one but without routing mechanism, and MIL-NCE* (* indicates that we used the same backbone as in our model). Correct video colored in green.

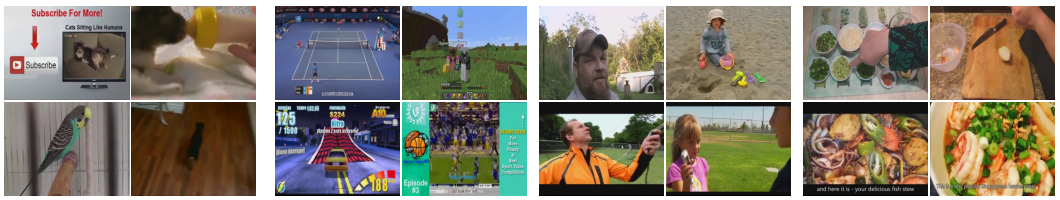

Figure 4: Top-4 videos with the highest activation for the particular capsule on the MSR-VTT dataset. Labels: #number of capsule: assumed learned "concept".

Retrieval Results We present retrieval results for three models - our self-attention based routing method, our approach without routing, and MIL-NCE - in Figure 3. Each column consists of the top-3 predictions for the given text query. Generally, routing achieves strong performance and retrieves visually varied videos; on the other hand, MIL-NCE tends to focus on specific objects or low-level visual cues leading to visually similar retrievals. In the first example, MIL-NCE retrieves videos of "melt butter", but the butter is melted in a pan and not an "oven". Notably, our approach successfully handles the extremely specific query "Put three rings of ketchup and two rings of mustard on the bottom bun" as shown in the second row. Additional qualitative results are presented in Appendix C.

What Individual Capsules Learn To further understand the entities or objects that are modeled, we examine the capsules' activations $\hat{p}_i^m$ (Equation 2) and show samples that have a high activation for a specific capsule. Ideally, if two samples have a high activation for the same capsule, then the entity that it represents should be present within both given inputs. To demonstrate this case, we select the videos in the downstream dataset MSR-VTT which lead to high activations for various capsules; we observe that different capsules model semantically distinct concepts as seen in Figure 4. The capsules learn to represent a wide range of entities: from general concepts like "games", "cooking", and "outdoor activities", to specific objects like "vegetables" and "cars". In Appendix C we include additional examples and show that these concepts are consistent across different datasets.

5 Conclusion

In this work, we proposed a novel multimodal capsule network that learns to model various entities within given modalities and maps them to a joint embedding space. To learn from a large amount of noisy video data, we present a scalable self-attention based capsule routing mechanism, which we show outperforms previous routing methods on this task. Furthermore, we find that the capsules are able to learn representations of various concepts and objects within each modality. Our comprehensive experimental evaluation demonstrates the effectiveness of our approach on two downstream zero-shot tasks on four datasets.

REPRODUCIBILITY STATEMENT

The source code for the experiments will be made available upon publication. We only use publicly available dataset and backbones.

REFERENCES

Karim Ahmed and Lorenzo Torresani. Star-caps: Capsule networks with straight-through attentive routing. In *Advances in Neural Information Processing Systems*, 2019.

Jean-Baptiste Alayrac, Adria Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In *Advances in Neural Information Processing Systems*, 2020.

Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *arXiv preprint arXiv:1911.12667*, 2019.

Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *Advances in Neural Information Processing Systems*, 2020.

Elad Amrani, Rami Ben-Ari, Daniel Rotman, and Alex Bronstein. Noise estimation using density estimation for self-supervised multimodal learning. *arXiv preprint arXiv:2003.03186*, 2020.

Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019.

Mohammad Taha Bahadori. Spectral capsule networks. *ICLR Workshop*, 2018.

Angie W Boggust, Kartik Audhkhasi, Dhiraj Joshi, David Harwath, Samuel Thomas, Rogério Schmidt Feris, Danny Gutfreund, Yang Zhang, Antonio Torralba, Michael Picheny, et al. Grounding spoken words in unlabeled video. In *CVPR Workshops*, 2019.

Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *Proceedings of the European Conference on Computer Vision*, 2014.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2009.

Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. Videocapsulenet: A simplified network for action detection. *arXiv preprint arXiv:1805.08162*, 2018.

Marzieh Edraki, Nazanin Rahnavard, and Mubarak Shah. Subspace capsule network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *Proceedings of the European Conference on Computer Vision*, 2020.

Taeyoung Hahn, Myeongjang Pyeon, and Gunhee Kim. Self-routing capsule networks. In *Advances in Neural Information Processing Systems*, 2019.

Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

David Harwath, Adria Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James Glass. Jointly discovering visual objects and spoken words from raw sensory input. In *Proceedings of the European Conference on Computer Vision*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *Proceedings of the International conference on artificial neural networks*, 2011.

Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. In *International conference on learning representations*, 2018.

Gabriel Ilharco, Yuan Zhang, and Jason Baldridge. Large-scale representation learning from visually grounded untranscribed speech. *arXiv preprint arXiv:1909.08782*, 2019.

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

Adam R Kosiorek, Sara Sabour, Yee Whye Teh, Geoffrey E Hinton, and Miss Jo STAFFORD-TOLLEY. Stacked capsule autoencoders. *Advances in Neural Information Processing Systems*, 2019.

Hilde Kuehne, Ahsan Iqbal, Alexander Richard, and Juergen Gall. Mining youtube-a dataset for learning fine-grained action concepts from webly supervised video data. *arXiv preprint arXiv:1906.01012*, 2019.

Rodney LaLonde, Ziyue Xu, Ismail Irmakci, Sanjay Jain, and Ulas Bagci. Capsules for biomedical image segmentation. *Medical image analysis*, 2021.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the International Conference on Machine Learning*, 2019.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. *arXiv preprint arXiv:2102.06183*, 2021.

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. Univilm: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020.

Vittorio Mazzia, Francesco Salvetti, and Marcello Chiaberge. Efficient-capsnet: Capsule network with self-attention routing. *arXiv preprint arXiv:2101.12491*, 2021.

Bruce McIntosh, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. Visual-textual capsule routing for text-based video segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2020.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2020.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, João Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. *arXiv preprint arXiv:2010.02824*, 2020.

Andrew Rouditchenko, Angie Boggust, David Harwath, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Rogerio Feris, Brian Kingsbury, Michael Picheny, Antonio Torralba, et al. Avlnet: Learning audio-visual language representations from instructional videos. In *Interspeech*, 2021.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829*, 2017.

Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. How2: a large-scale dataset for multimodal language understanding. *arXiv preprint arXiv:1811.00347*, 2018.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Aisha Urooj Khan, Hilde Kuehne, Kevin Duarte, Chuang Gan, Niels Lobo, and Mubarak Shah. Found a reason for me? weakly-supervised grounded visual question answering using capsules. *arXiv e-prints*, 2021.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Min Yang, Wei Zhao, Jianbo Ye, Zeyang Lei, Zhou Zhao, and Soufei Zhang. Investigating capsule networks with dynamic routing for text classification. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018.

Suofei Zhang, Quan Zhou, and Xiaofu Wu. Fast dynamic routing based on weighted kernel density estimation. In *Proceedings of the International Symposium on Artificial Intelligence and Robotics*, 2018.

Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3d point capsule networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

A Additional Ablations

Shared weights For our proposed architecture, we share weights across the various modalities after the initial capsules are extracted. Not only does this reduce the number of learned parameters for the network, but we find that it leads to learning improved representations. We present results in Table 6. Generally, for data more closely related to the training data (for example, evaluating on YouCook2) the use of shared weights leads to improved performance.

Table 6: Evaluation using shared weights

	YouCook2				MSR-VTT			
Modalities	R@1	R@5	R@10	Med. R	R@1	R@5	R@10	Med. R
Not Shared	16.8	35.4	44.6	15	9.5	22.8	30.3	30.5
Shared	19.3	37.8	47.3	13	9.3	21.4	30.9	37

B Self-attention Architectural Details

For our self-attention routing procedure we first use linear projections to generate the query-key-value. Given that there are C input capsules and the output capsules have dimension d_2, the query, key, and value matrices have shape $Q, K, V \in \mathbb{R}^{C \times d_2}$. The output of the multi-head self-attention operation,

$$V' = \text{softmax}\left(\frac{QK^T}{\sqrt{d_2}}\right)V, \tag{5}$$

is a matrix of the same dimension. We then apply normalization across the columns (i.e. capsule feature dimension) as well as two fully connected linear layers, and dropout, with hidden dimension 1024 and output dimension d_2. A residual connection from V' to the output capsule features, followed by normalization across the capsule feature dimensions.

C Additional Qualitative Results

Retrieval Quality In the case of retrieval, we show three text queries together with their three closest video representations in Figure 5. It becomes clear that all video representations show a close match for the described scene. Additionally, one has to remark that the retrieved video examples for each query do show sufficient variance with respect to color, view point, and other low-level cues. This can be seen as an indicator that the learned clustering is based on some high-level common concepts rather than on the pure co-occurrence of low-level feature representations.

Retrieval Results We present additional retrieval results for three models - our self-attention based routing method, our approach without routing, and MIL-NCE - in Figure 6. Each column consists of the top-3 predictions for the given text query. Generally, routing achieves strong performance and retrieves visually varied videos; on the other hand, MIL-NCE tends to focus on specific objects or low-level visual cues leading to visually similar retrievals. The first three rows consists of examples where our self-attention based routing correctly retrieves videos but the other two methods do not. For general queries, like "grill the ribs" and "flip the pancakes" in the bottom two rows, there are many relevant videos to choose from. Only the no-routing method obtains the "correct" video in its top-3 predictions, but these "failure cases" for our method and MIL-NCE would be considered correct retrievals by human standards.

Capsule Activations In Figure 7, we include videos which correspond to various capsules' highest activations across three different datasets: HowTo100M, MSR-VTT, and YouCook2. The concepts tend to remain consistent across the different datasets. In the final three rows, we present examples where the concept does not exist within the target dataset: "repair", "games", and "pets/animals" are not present within the cooking dataset, YouCook2. Since these capsules learn to represent these specific concepts/entities, their highest activation corresponds to seemingly random videos.

Figure 5: Qualitative evaluation: examples of top-3 zero-shot text-to-video retrieval results on the YouCook2 dataset. Correct video colored in green.

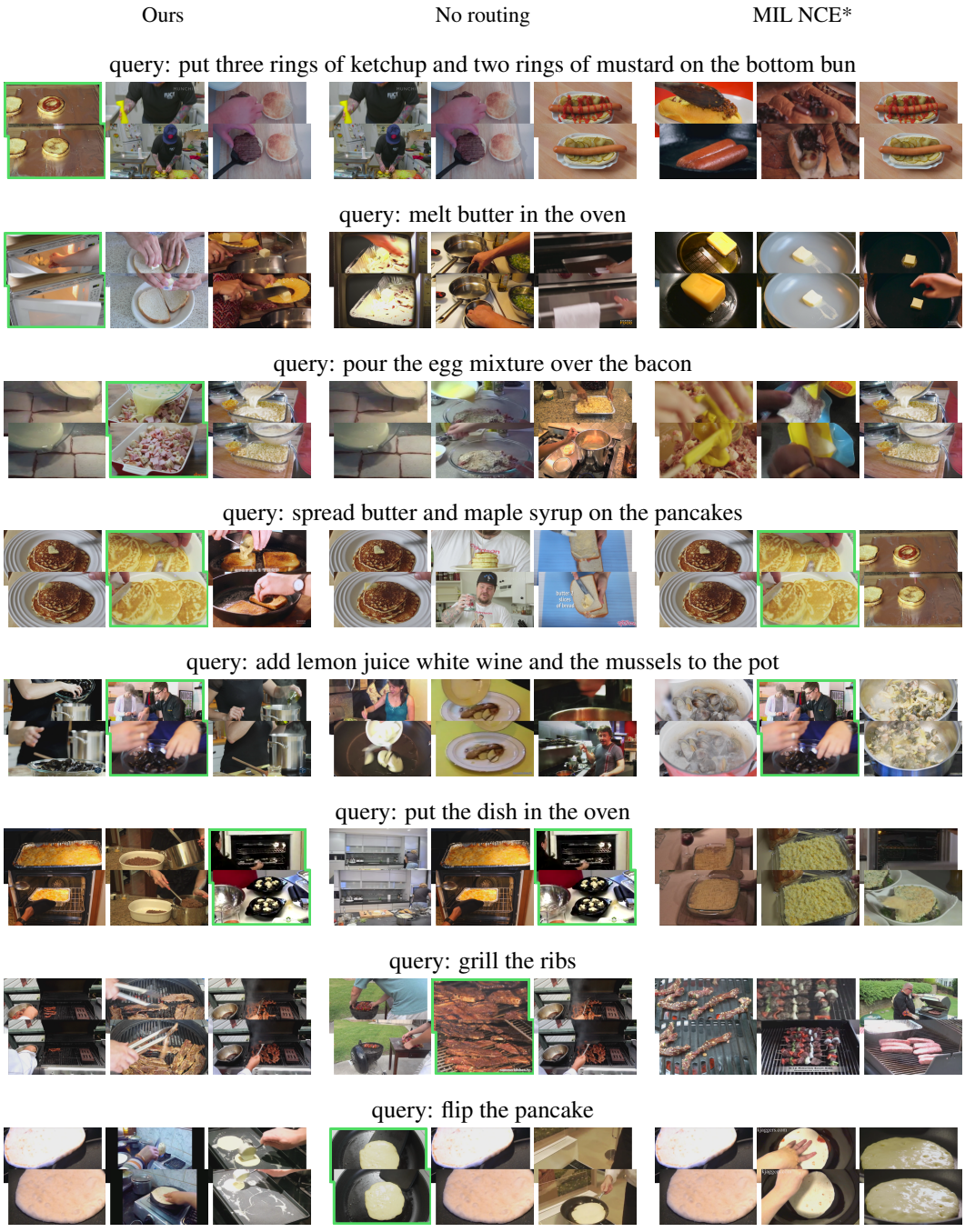

Figure 6: More qualitative examples: top-3 zero-shot text-to-video retrieval results on the YouCook2 dataset for the proposed approach with self-attention based routing, the same one but without routing mechanism, and MIL-NCE* (* indicates that we used the same backbone as in our model). Correct video colored in green.

Figure 7: Extended figure with examples of capsule highest activations: top-4 videos with the highest activation for the particular capsule for the HowTo100M, MSR-VTT, and YouCook2 datasets. Labels: #number of capsule: assumed learned "concept".