OmniVec: Learning robust representations with cross modal sharing

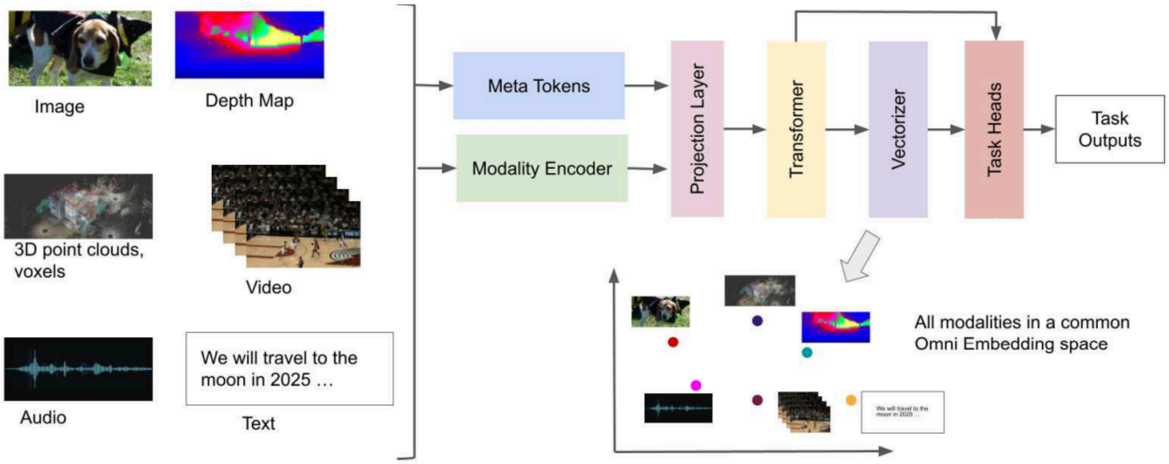
Key Ideas

Motivation:

Introducing a unified network that works with multiple modalities like image, video, audio etc.

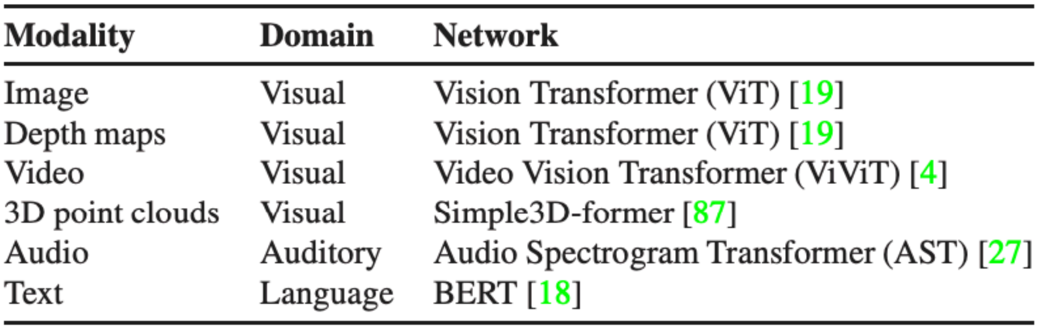
- Highlight that learning tasks together with a unified network can lead to regularization effects, as a large amounts of shared parameters are trained to perform varied tasks, and hence are more likely to extract meaningful representations from data without overfitting to one task or modality.
- It can also aid in utilizing available labelled data from different domains, hence potentially eliminating the cost and effort of labelling large amounts of data in a specific modality for a specific task.

Architecture:



The architecture have the following components:

Modality encoders: These encoders are modality specific and could be any deep network that encodes the raw data into some feature representation. In the paper they have used the following modality encoders for different modalities.



Modality	Domain	Network
Image	Visual	Vision Transformer (ViT) [19]
Depth maps	Visual	Vision Transformer (ViT) [19]
Video	Visual	Video Vision Transformer (ViViT) [4]
3D point clouds	Visual	Simple3D-former [87]
Audio	Auditory	Audio Spectrogram Transformer (AST) [27]
Text	Language	BERT [18]

Meta tokens: Meta token is a vector extracted from any given modality. It represents the modality by encoding things like modality (I), size of temporal dimension (T), height (H), width (W) in spatial dimension, number of channels (C) and length or number of tokens (L).

- For image modality, the meta token could look like [T:1, H: height of the patch, W: width of the patch, C:3 (RGB), L:1 (non relevant)

Projection Layer: The projection layer takes the meta token and output of the modality encoder as inputs. The output of the modality encoder should be represented as patches. A linear projection is applied to these patches and they now become a n-dimensional vector. The meta-tokens discussed above come into play here. They condition the projection layer with respect to a given modality.

Each modality will have the learnable weight like this. This is known by the help of the meta-tokens.

$$W_{ip} \in \mathbb{R}^{t \cdot h \cdot w \cdot c \cdot l \times n}$$

Transformer: This is a common bottleneck for all the modalities and takes the output patches of the projection layer for all the modalities as input and output the features.

Vectorizer: Vectorizer component takes the patches from the transformer as input, concatenates them and pass them through a linear mapper to get a d-dimensional output for a given input. Note that the vectorizer step is optional.

Task Heads: Task heads are the independent task specific networks that take the output from the transformer block or vectorizer and performs tasks like classification, semantic segmentation etc.

Training Methodology:

They perform the model training in two steps:

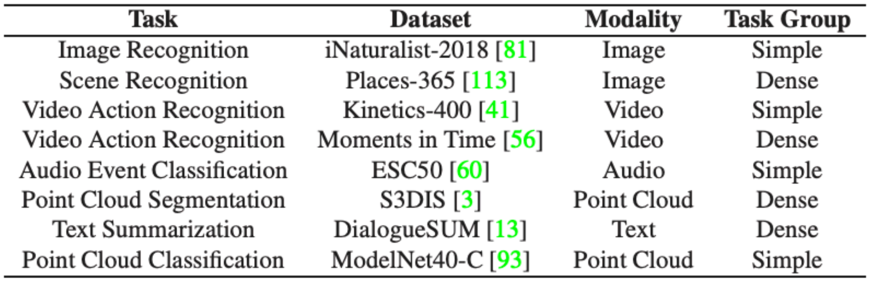
- Masked pre-training
- Multimodality, Multi-dataset, task-specific end-to-end training

Masked pre-training: For masked pre-training, the general approach in Masked Auto-encoder pre-training is used. Here the pre-training is performed across multiple datasets of different modalities. At this point the task head is irrelevant as we use a encoder-decoder MAE network to do the pretraining. While pretraining, the minibatch contains patches from multiple modalities. This ensures the cross modal interaction between the samples.

They use AudioSet (audio), Something-Something v2 (SSv2)(video) , English Wikipedia (text), ImageNet1K(image), SUN RGB-D (depth maps), ModelNet40 (3D point cloud) for pretraining the network.

End-End training: In the second part of the training, they use the pre-trained model from the Masked pre-training stage and fine-tune it end to end using the task-heads. There could be n-task heads based on the modality and task.

They also introduce something called task grouping. They group all the task into two groups. Simple and Dense. The table below outlines these tasks with respect to the different datasets and modalities.



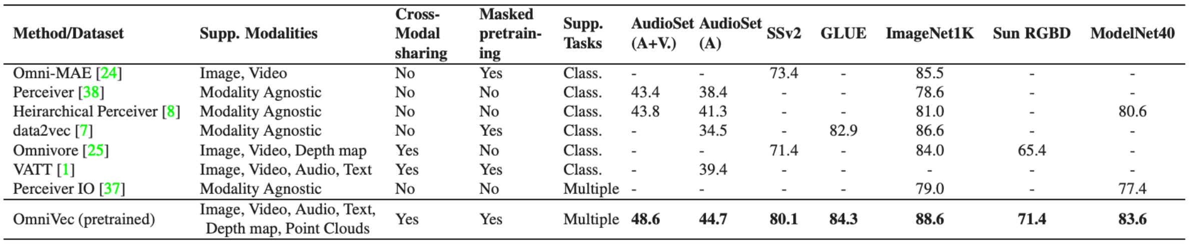
Task	Dataset	Modality	Task Group
Image Recognition	iNaturalist-2018 [81]	Image	Simple
Scene Recognition	Places-365 [113]	Image	Dense
Video Action Recognition	Kinetics-400 [41]	Video	Simple
Video Action Recognition	Moments in Time [56]	Video	Dense
Audio Event Classification	ESC50 [60]	Audio	Simple
Point Cloud Segmentation	S3DIS [3]	Point Cloud	Dense
Text Summarization	DialogueSUM [13]	Text	Dense
Point Cloud Classification	ModelNet40-C [93]	Point Cloud	Simple

The idea is, the fine-tuning on all these datasets by alternating the training epochs between the simple and dense tasks.

For end-end training, they use the datasets mentioned in the above table.

Impressions

Comparison of pretrained OmniVec with similar methods:



Method/Dataset	Supp. Modalities	Cross-Modal sharing	Masked pretrain-ing	Supp. Tasks	AudioSet (A+V.)	AudioSet (A)	SSv2	GLUE	ImageNet1K	Sun RGBD	ModelNet40
Omni-MAE [24]	Image, Video	No	Yes	Class.	-	-	73.4	-	85.5	-	-
Perceiver [38]	Modality Agnostic	No	No	Class.	43.4	38.4	-	-	78.6	-	-
Heirarchical Perceiver [8]	Modality Agnostic	No	No	Class.	43.8	41.3	-	-	81.0	-	80.6
data2vec [7]	Modality Agnostic	No	Yes	Class.	-	34.5	-	82.9	86.6	-	-
Omnivore [25]	Image, Video, Depth map	Yes	No	Class.	-	-	71.4	-	84.0	65.4	-
VATT [1]	Image, Video, Audio, Text	Yes	Yes	Class.	-	39.4	-	-	-	-	-
Perceiver IO [37]	Modality Agnostic	No	No	Multiple	-	-	-	-	79.0	-	77.4
OmniVec (pretrained)	Image, Video, Audio, Text, Depth map, Point Clouds	Yes	Yes	Multiple	**48.6**	**44.7**	**80.1**	**84.3**	**88.6**	**71.4**	**83.6**

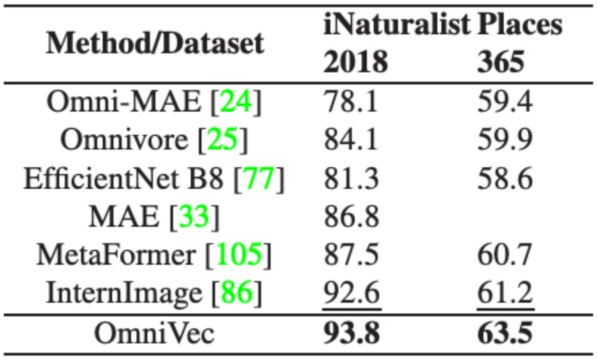
Table 2. **Comparison of OmniVec framework with similar methods that work on multiple modalities**. We compare OmniVec with masked pretraining with the best reported results from respective publications of the compared methods. Supp. Tasks and Supp. Modalities indicate Supported Tasks and Supported Modalities by respective networks. In Supported (Supp.) Tasks, Class. indicates classification.

Here, pre-trained means the model that was pre-trained using MAE and then end-end fine-tuned with task heads trained with modality mixing and target grouping on datasets mentioned in section above. Note that the datasets used for pre-training and fine-tuning are different.

They show that the Omni-vec outperforms all the other multimodal methods on the given datasets.

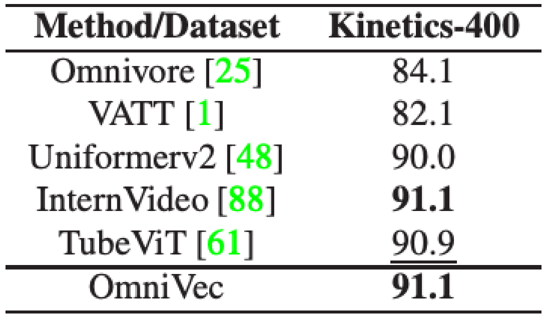
Image Modality:

Omni-Vec shows to outperform the other modality agnositic approaches on the image classification tasks by a good margin



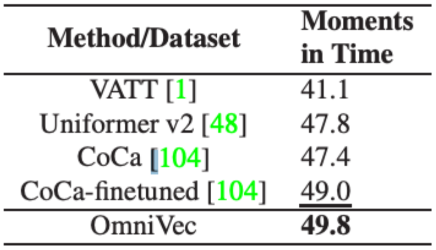
Method/Dataset	iNaturalist 2018	Places 365
Omni-MAE [24]	78.1	59.4
Omnivore [25]	84.1	59.9
EfficientNet B8 [77]	81.3	58.6
MAE [33]	86.8	
MetaFormer [105]	87.5	60.7
InternImage [86]	92.6	61.2
OmniVec	**93.8**	**63.5**

Video Modality:



Method/Dataset	Kinetics-400
Omnivore [25]	84.1
VATT [1]	82.1
Uniformerv2 [48]	90.0
InternVideo [88]	**91.1**
TubeViT [61]	90.9
OmniVec	**91.1**

Table 4. **Kinetics-400** top-1 accuracy.

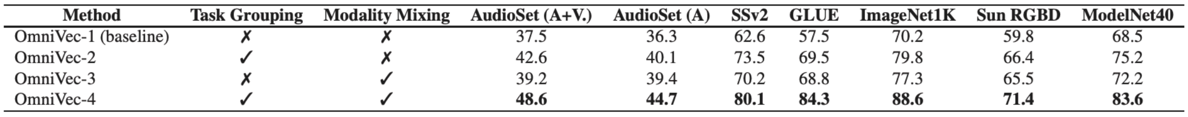


Method/Dataset	Moments in Time
VATT [1]	41.1
Uniformer v2 [48]	47.8
CoCa [104]	47.4
CoCa-finetuned [104]	49.0
OmniVec	**49.8**

Table 5. **Moments in time** top-1 accuracy.

Ablation Studies:

Impact of task grouping and modality mixing:



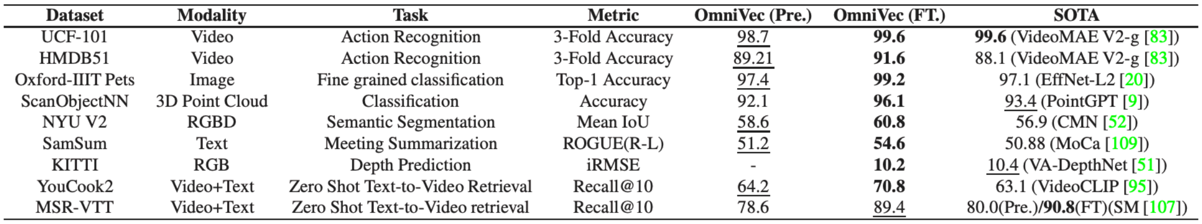
Method	Task Grouping	Modality Mixing	AudioSet (A+V.)	AudioSet (A)	SSv2	GLUE	ImageNet1K	Sun RGBD	ModelNet40
OmniVec-1 (baseline)	✗	✗	37.5	36.3	62.6	57.5	70.2	59.8	68.5
OmniVec-2	✓	✗	42.6	40.1	73.5	69.5	79.8	66.4	75.2
OmniVec-3	✗	✓	39.2	39.4	70.2	68.8	77.3	65.5	72.2
OmniVec-4	✓	✓	**48.6**	**44.7**	**80.1**	**84.3**	**88.6**	**71.4**	**83.6**

It shows that both task grouping and modality mixing individually/ collectively help with the model performance.

Generalization on unseen datasets:

The Omni-Vec model when evaluated on these unseen datasets, performs on par or better than the stat of the art models.

Image classification (Oxford-IIIT Pets), Video Classification (UCF-101 , HMDB51), 3D point cloud classification (ScanObjectNN), 3D point cloud segmentation (NYUv2) and text summarization (SamSum)



Dataset	Modality	Task	Metric	OmniVec (Pre.)	OmniVec (FT.)	SOTA
UCF-101	Video	Action Recognition	3-Fold Accuracy	98.7	**99.6**	**99.6** (VideoMAE V2-g [83])
HMDB51	Video	Action Recognition	3-Fold Accuracy	89.21	**91.6**	88.1 (VideoMAE V2-g [83])
Oxford-IIIT Pets	Image	Fine grained classification	Top-1 Accuracy	97.4	**99.2**	97.1 (EffNet-L2 [20])
ScanObjectNN	3D Point Cloud	Classification	Accuracy	92.1	**96.1**	93.4 (PointGPT [9])
NYU V2	RGBD	Semantic Segmentation	Mean IoU	58.6	**60.8**	56.9 (CMN [52])
SamSum	Text	Meeting Summarization	ROGUE(R-L)	51.2	**54.6**	50.88 (MoCa [109])
KITTI	RGB	Depth Prediction	iRMSE	-	**10.2**	10.4 (VA-DepthNet [51])
YouCook2	Video+Text	Zero Shot Text-to-Video Retrieval	Recall@10	64.2	**70.8**	63.1 (VideoCLIP [95])
MSR-VTT	Video+Text	Zero Shot Text-to-Video retrieval	Recall@10	78.6	89.4	80.0(Pre.)/**90.8**(FT)(SM [107])