CS2952N Final Project Report: Shared Uniform Space (SUS) Embedding

Raymond Del Vecchio, David Lubawski, Nitya Thakkar

May 12, 2023

1 Introduction

For most tasks, current state of the art deep learning solutions to computational problems involve training a single model, resulting in one set of weights highly specified for that task. There are some drawbacks to this. With one set of weights per task, the model may overfit to the task rather than learning abstractly about the task itself. In ImageNet [4], for example, a model may learn that the presence of tires and a windshield implies a car rather than learning some representation of what a car *is*. Further, since these weights are highly specialized only for one task (image classification, reconstruction, captioning, etc), the potential for transfer learning to other tasks is limited. Theoretically, a model that is good at image classification could also help inform the image reconstruction task with its learned knowledge about images and the objects contained within them. However, since a model's weights are highly tuned for one specific task, combined with the learning vs task completion problem detailed before, researchers have not been able to use this potential.

To assist this problem, we propose the use of a shared embedding space for multiple related and unrelated tasks. We will train this shared embedding space as an intermediate layer for all target tasks, and will remain constant between them. Thus, if our models are able to perform highly on each task, we can be confident that this shared embedding space contains a large amount of relevant information about each task, rather than overfitting to each specific task itself. This shared embedding space is also used to enhance the quality of the model on each task, pulling from its representation and knowledge of other tasks to inform a target task. In this report, we introduce a Shared Uniform Space (SUS) Embedding framework to train such an embedding space for multiple tasks, incorporating multiple modalities into the embedding space, and applying the SUS to complete each target task.

Our code is publicly available at
https://github.com/nityathakkar/sus_final_project.git.

2 Related Works

A single weights space to perform multiple tasks is commonly referred to as a "generalist agent". There is much literature on the creation and training of generalist agents. A very early attempt was UberNet [1]. UberNet was trained to complete multiple vision tasks, ranging from low to high level, such as boundary detection, semantic segmentation, and object detection, with a single Convolutional Neural Network. Tasks and data were collected from a variety of sources, and the training dataset was formed by the "set union" of all these tasks. UberNet achieved near-SOTA accuracy on all tasks applied on, with only small decline in performance when generating other tasks simultaneously, showing how generalist agents can be highly powerful "swiss knife" solutions to related tasks.

In 2022, DeepMind introduced Gato [3]. Gato is a generalist agent tasked with performing a large quantity of unrelated gaming, image processing, NLP, and robotic tasks. At inference time, data passed into Gato is pre-processed and tagged to mark the specific task the model must complete. Although ultimately one large set of weights, Gato has faced criticism that the model simply learns multiple sub-networks, or subspaces of the weights for each task, due to this pre-process tagging, rather than drawing predictions from the entire weight space as a whole.

More recently, a paper [2] published from Google Research presents a framework for training multitask agents via offline reinforcement learning. An RL policy is trained on multiple related Atari game tasks simultaneously. The researchers were then able to fine-tune this policy to other game tasks, drawing on the pre-trained multitask ability. They found that such multitask offline pre-training was able to significantly boost performance of the model upon fine-tuning in both an online and offline setting. Thus, in these papers, we see the large potential of multitask embedding spaces, representations, and models for higher and more generalizable task performance.

3 Methods

3.1 Data

We train using the cifar10 dataset for our images and the WikiText dataset for our text dataset. For images, we subset the cifar10 dataset to use 20,000 random training examples for each training image task. For text, we randomly subset 10,000 sequences from the WikiText dataset and further slice each 150 token sequence into 10 by 15 token sequences.

3.2 Model Architecture

We formulate our model in 3 parts: a modality cap, the SUS embedding space and the task-specific cap (see Figure 1). Modality caps are specific to each input modality (ex. Text, Image, etc.) and at inference time are ignorant of the task trying to be achieved during inference. The modality caps are kept the same between tasks, that is, during training all image examples are passed through the same modality cap. During our training, our image modality cap architecture was comprised of 2 CNN layers and a Dense layer. Our text modality cap used an embedding layer with 24 dimensions, a transformer encoder block with 4 attention heads followed by a dense layer. These encoder architectures where originally simply MLPs but we found that we could use fewer parameters and achieve better performance with these encoding schemes.

The SUS embedding space was an MLP consisting of 3 dense layers with LeakyReLU activations. The Embedding takes in a vector of some size and compresses it down to a latent size before passing it on to the task cap. In our experiments, we use an input size of 2048 and output of 512. Finally, each particular task had its own task cap which was a one layer perceptron for image classification and next token prediction and a 2 layer MLP for image reconstruction.

When training on multiple tasks at once, we randomly take a batch of input examples from a given task at each train step. At inference time, the modality and task of the inference example dictate which caps to use. The model then assembles the caps and embedding used for that particular forward pass.

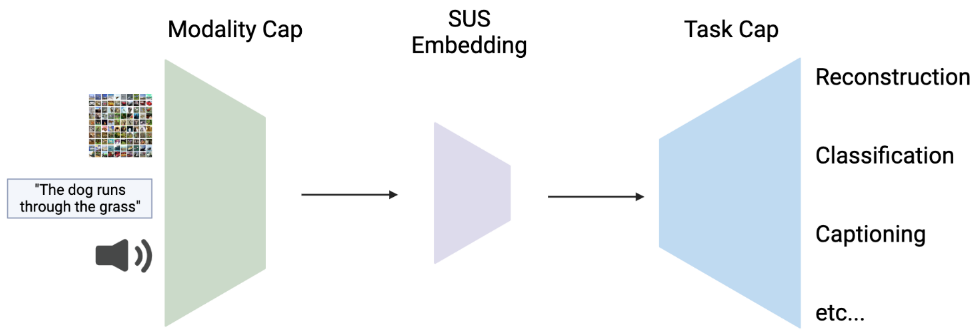


Figure 1: Overview of our model: a modality cap, the SUS embedding space and the task-specific cap

4 Experiments

For the three tasks (image classification, image reconstruction, and next word prediction), we ran five models total: train on each task individually, train on the two image tasks together, and train on all three together. For the image classification task, we report classification accuracy. For the image reconstruction task, we report mean-squared error (MSE) loss. For the next word prediction task, we report perplexity. All the metrics reported are on the validation set. Are results are summarized in Table 1

Training Task	Accuracy	MSE	Perplexity
Image classification	**0.495**	-	-
Image reconstruction	-	**77.09**	-
Next word prediction	-	-	**1.31**
Image classification + reconstruction	0.455	78.71	-
All 3 tasks	0.459	84.07	3.38

Table 1: Results from training the model on the various combinations: each task alone, the two image tasks together, and all 3 tasks together. We find that the next word prediction task performs best alone, and we expect performance would improve with more data and compute. The image tasks perform comparably when the unrelated text task is added. Accuracy is for image classification, MSE loss is for image reconstruction, and perplexity is for text generation.

We also produce visualizations for the image reconstruction task to understand how well the model is reconstructing the images from the latent space. The results are summarized in figure 2

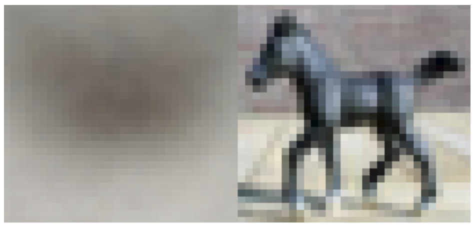
Overall, our results indicate that the SUS model is able to perform well on multiple tasks when trained together. This performance indicates it learns high-level representations for the different modalities, rather than task-specific representations.

5 Conclusion

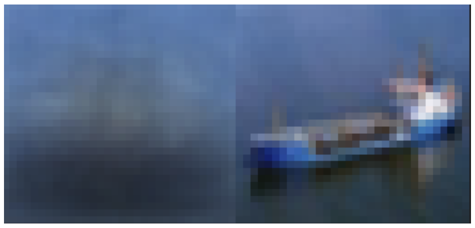
In conclusion, our SUS model is indeed able to produce relatively accurate results across each task when trained jointly on all of them. However, it is extremely computationally expensive, and so we would need to re-run these experiments with more data and on more compute to truly validate the results. Based on our results, we found that the model does indeed seem to learn high-level representations for the different modalities, rather than task-specific representations. Additionally, we hypothesize that the potential for the model to be good at all three tasks is likely limited by the similarity of the two image tasks and the dissimilarity of the third (text) task. We expect by training on more tasks that involve text – and even additional data modalities – we can learn a richer embedding space.

In the future, we also hope to conduct an ablation study to understand if the SUS embedding layer is truly necessary. To do so we would remove the SUS embedding layer and simply train the caps, and then analyze the results when training on multiple tasks at the same time. We also want to analyze the SUS embedding space to understand how the different modalities and tasks may cluster. We plan to do so using principal component analysis. Overall, we were able to train a multi-modal embedding space that is resistant to subnetworks because we lack direct prompting and also have a specialized compression scheme.

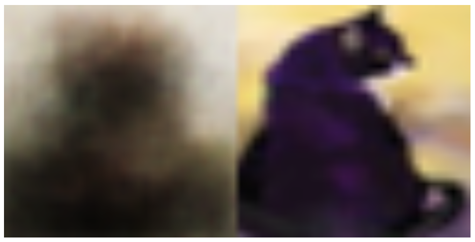
Ultimately, our project serves as verifying the SUS proof-of-concept for training and employing a unified embedding space across many tasks, and future work can build upon the technique with more compute power, more data, and novel tasks.



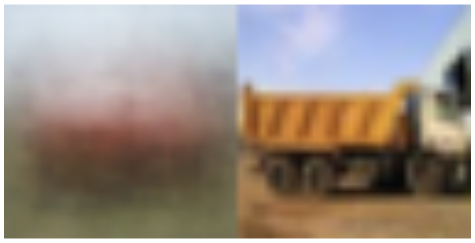
(a) Example image when training on image reconstruction alone



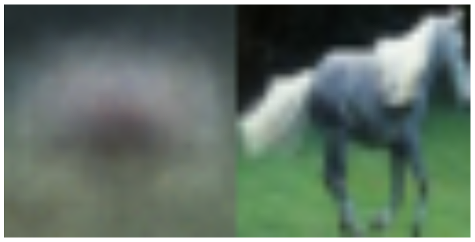
(b) Example image when training on image reconstruction alone



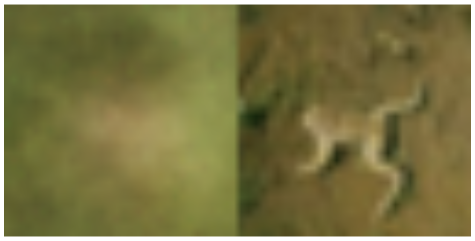
(c) Example image when training on image reconstruction and classification together



(d) Example image when training on image reconstruction and classification together



(e) Example image when training on all tasks at the same time



(f) Example image when training on all tasks at the same time

Figure 2: Sample testing images from the reconstruction process. We found that, while the loss may increase as the embedding space is trained on more tasks, the reconstructions produced do not significantly decrease in quality when manually observed. We found the reconstructions when trained on both image tasks together and all three tasks to be visually better than the results when only trained on reconstruction.

6 Division of Labor

Our division of labor was highly equal in nature. Each student brainstormed ideas together in person, implemented the project on VSCode live share together, put heads together to debug when issues arose, did research / wrote parts of the report live on Overleaf together, and contributed different slides to our presentation. Each student is satisfied with the total work put in by themselves and each other student. There was no case in which one student worked on parts of the project where others did not.

References

[1] Iasonas Kokkinos. Ubernet: Training a 'universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory, 2016.

[2] Aviral Kumar and Sergey Levine. Pre-training generalist agents using offline reinforcement learning, 2022.

[3] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent, 2022.

[4] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015.