POINTSPIDER: Learning linear mappings for multimodal alignment of images and point clouds

Vijay Jaisankar, Jaya Sreevalsan Nair

***Abstract*— In this paper, we propose POINTSPIDER, a linear mapping model bridging embeddings of point clouds and images. We explore minimal test dataset creation through Double Roulette sampling and evaluate the efficacy of pretrained Large Image models in zero-shot classification. By using pre-trained Pointnet and CLIP-ViT backbones, we obtain a validation Mean L1 loss value of $\approx$ 0.25 using linear layers with layer normalisation.**

I. INTRODUCTION

A. Multimodal learning

Multimodal learning (MML) is a general approach to building AI models that can extract and relate information from multimodal data [1]. MML systems are being increasingly adopted, owing to their correspondence to human perception. Recent pathbreaking developments in MML systems like OpenAI's GPT-4 [2] and Google Deepmind's Gemini [3] have demonstrated considerable academic and commercial value for such systems. In this regard, research into MML systems for aligning different modalities is a valuable contribution to understanding the outputs of different sensors and inputs into systems.

B. Point clouds as input modalities to MMLs

There has been considerable work done on learning generalisable representations for point clouds conjoined with various other modalities.

For example, ULIP [4] utilises large pre-trained image and text encoders and uses a sample of the respective data pool to align the corresponding 3D point clouds.

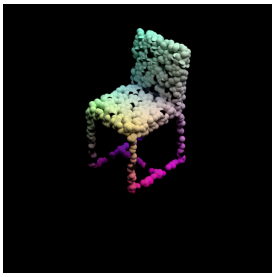
We wish to achieve a similar multimodal alignment scheme on a *smaller scale*. We hypothesise that by using a small set of high-quality inputs and a lightweight model architecture, the embeddings produced through multimodal alignment can be used for effective representations, and can also then be fine-tuned to downstream tasks while using relatively lower computational resources.

C. Overview of POINTSPIDER

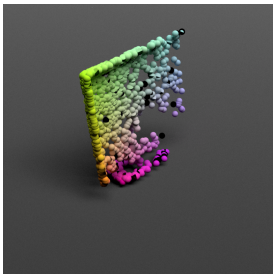
Building on I-B, we propose POINTSPIDER, a Deep Neural Network (DNN) [5] architecture that aligns point clouds and their corresponding images. By keeping the mapping network relatively lightweight, we hypothesise that it can be further finetuned for future use cases.

II. DATASET

For our work, we use the ModelNet-40C [6] dataset, which consists of 40 classes of models stored with various corruptions, like shearing and occlusion. The advantages of using this dataset for our task are two-fold:



(a) Visualisation of a *clean* point cloud of a chair



(b) Visualisation of a *corrupted* point cloud of a monitor

Fig. 1: Samples of the objects in the Modelnet40-C dataset

- The point clouds are of comparatively lesser complexity and size, hence enabling faster training
- The various corruptions can serve as strong tests for the embeddings and conversely, a model trained on this dataset has notable adversarial resistance to the same.

Figure 1 contains a few samples from the ModelNet40-C dataset. The corresponding visualisations were generated by using the procedure detailed in III.

III. GENERATING VISUALISATIONS FOR POINT CLOUDS

To align the modalities of images and point clouds, we use the Mitsuba renderer [7], as provided in the public *PointVizualisation* repositoty on Github [8]. To ensure consistency across the experiments, we only consider the inputs with *1024* points per cloud. As per Table I, we note that, on the NVIDIA GeForce GTX 1050 GPU, the *cuda-rgb* variant is $\approx$3x faster than *llvm-rgb* and advise future researchers to prefer *llvm-rgb* over *scalar-rgb* in the absence of a GPU.

TABLE I: Performance of Mitsuba renderer variants

Mitsuba variant name	PPS (points per second)
scalar-rgb	14.625
llvm-rgb	18.720
cuda-rgb	**56.161**

IV. BUILDING ON REPRESENTATIONS

A. Zero-shot classification

Zero-shot classification [9] is a powerful paradigm in machine learning, wherein by learning from auxillary data, the models can classify new inference data whose classes are disjoint from those of the auxillary data. Zero-shot classifiers are pre-trained on large amounts of data, making

their embeddings contrastively expressive.

Figure 2 shows a schematic diagram of the Contrastive language-image pretraining(CLIP) paradigm of matching images with their corresponding captions. We derive from the immense expressive power of such models by the virtue of the diversity and size of their pre-training dataset.

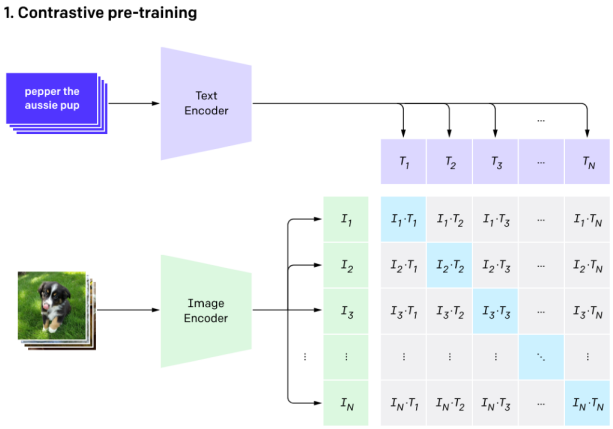


Fig. 2: An overview of the CLIP [10] paradigm

B. Double Roulette Sampling

To generate a diverse minimal testing dataset to evaluate candidate zero-shot classifier, we propose *Double Roulette Sampling*. As alluded to in III, we limit our search for candidate point clouds to those who represent each object with 1024 points - this results in 55 corruption variants to choose from.

Notation

- C refers to the set of classes of objects - In ModelNet-40C, $|C| = 40$.
- For each class index i, D_i consists of the list of objects in ModelNet-40C belonging to that class.
- T refers to the set of transformations/corruptions present in the ModelNet-40C. In particular, for a given input object of class i, there will be $|T|$ entries in D_i - in our experiments, $|T| = 55$.

Algorithm 1 Double Roulette Sampling

Input: Class indices indexed from 1 to $|C|$
Input: List of objects represented by D_1 to $D_{|C|}$
Output: $|C|$ point clouds that constitute the test set for evaluating zero-shot classifiers
Initialisation :
$O = \Phi$
for $classIndex = 1$ to $|C|$ **do**
  $chosenCloud = RandomSample(D_{classIndex})$
  $O = O \bigcup \{chosenCloud\}$
end for
return O

Algorithm 1 contains the selection procedure for the minimal test set. At the end of this procedure, we have 40 diverse and unique samples, each belonging to a unique category of objects.

C. Evaluating Candidate models for Image Embeddings

For this experiment, we consider the following models for the purposes of the zero-shot classification task:

- clip-vit-large-patch14 [1]
- clip-vit-base-patch32 [2]
- CLIP-ViT-H-14-laion2B-s32B-b79K [3]
- CLIP-ViT-L-14-DataComp.XL-s13B-b90K [4]

For each model, we feed in the point cloud visualiations of the minimal test set along with a set of 40 ordered prompts of the form *A model of* ***objectType***. We evaluate the performance of these models through the top-k accuracy scores, where $k = \{1, 3\}$.

TABLE II: Performance of zero-shot classifiers over the minimal test set

Model name	Top-1 acc.	*Top-3 acc.*
clip-vit-large-patch14	0.125	0.150
clip-vit-base-patch32	0.125	0.200
CLIP-ViT-H-14-laion2B-s32B-b79K	**0.300**	**0.400**
CLIP-ViT-L-14-DataComp.XL-s13B-b90K	0.225	0.350

In lieu of the results summarised in Table II, we use *CLIP-ViT-L-14-DataComp.XL-s13B-b90K* as the source for image embeddings.

D. Point Cloud Embeddings

For embedding point clouds, we use Pointnet [11] through a trained checkpoint for the ModelNet-40C dataset. We use the global feature vector block for the point cloud as a whole and leverage the adversarial robustness capabilities of this model.

V. Multimodal alignment through feed-forward networks

In order to integrate the robustness of the Pointnet encoder and the expressiveness of CLIP-ViT, we learn a *linear mapping* from the embeddings of the point cloud to those of the images. We note that the efficacy of such a network has been demonstrated in the image-text alignment setting [12], which motivates our use case.

A. Model architecture

POINTSPIDER consists of the following blocks:

- 3 Upsampling layers that convert the input embedding into higher-dimensional vectors
- 3 Downsampling layers that convert the higher-dimensional vectors back into the dimensions of the target output embedding

Each Upsampling layer progressively doubles the current embedding size, and each Downsampling layer progressively halves the current embedding size. Both Pointnet's global

[1]https://huggingface.co/openai/clip-vit-large-patch14
[2]https://huggingface.co/openai/clip-vit-base-patch32
[3]laion/CLIP-ViT-H-14-laion2B-s32B-b79K
[4]https://huggingface.co/laion/CLIP-ViT-L-14-DataComp.XL-s13B-b90K

feature and CLIP-ViT produce 1024 dimensional embeddings, hence enabling this symmetrical structure.

In addition to this *BASE* architecture, we introduce two new variants of *POINTSPIDER*, by applying LayerNorm [5] and LocalResponseNorm [6] (with 1 neighbouring channel) into all blocks except for the last downsampling layer.
We denote these new variants *LNORM* and *RNORM* respectively.

B. Experimental setup

We divide the ModelNet-40 dataset randomly into 2048 train point clouds and 420 test point clouds. For each of these point clouds, we generate corresponding images, as described in III.
We train these models for 1000 epochs using the Adam [13] optimiser with a learning rate of 0.005, using the *Mean L1 loss* between the transformed Pointnet embeddings and the corresponding CLIP-ViT embeddings.

C. Results

We report the performance of each variant of POINTSPIDER as a function of the Mean L1 loss on the test section of the dataset. As a lower loss value is more optimal, we report the L_1DIV scores, defined as follows:

$$L_1DIV = \frac{1}{Mean\_L1\_Loss}$$

TABLE III: Performance of POINTSPIDER variants over the ModelNet-40 test set

Model variant	L_1DIV **score**
BASE	1.529
LNORM	**3.996**
RNORM	1.569

Based on the results in Table III, we conclude that using Layer Normalisation in the feed-forward mapping network is the most optimal setting. We also note the strong performance of this variant in the task and hence deem this model architecture promising for the task of multimodal alignment of point clouds and images.

VI. Discussions

A. Point cloud similarity measures

Such an architecture allows researchers to find similarities between two point clouds *without the need of projecting them to the image space*, saving up compute resources. On using the mapping block on top of the Pointnet outputs, the we hypothesise that the resultant embedding vectors capture both the expressiveness of the CLIP-ViT model and the geometric and adversarial understanding of the underlying Pointnet model.
We note the following areas of interest of the same:

- Removing duplicate point clouds from a dataset
- Finding similar point clouds to enable iterative editing

[5]https://pytorch.org/docs/stable/generated/torch.nn.LayerNorm.html
[6]https://pytorch.org/docs/stable/generated/torch.nn.LocalResponseNorm.html

B. Future Work

We note the following themes for future work for POINTSPIDER.

- Experimenting with different point cloud embedding architectures like PointMLP [14] and PointTransformer [15].
- Benchmarking and distilling the embeddings of large (possibly MML) models including Point Clouds as a modality, like ULIP and PointBERT [16].
- Fine-tuning the embeddings for other domains, for example, the automotive domain.

VII. Conclusion

In this paper, we have looked at the formulation of POINTSPIDER, a linear network for multimodal alignment of point clouds and images. We have also looked at the potential use cases of such a system and also note some interesting future directions for this project.

VIII. Acknowledgement

We would like to acknowledge the following Github repositories for their valuable implementations of crucial sections of POINTSPIDER:

- https://github.com/jiachens/ModelNet40-C
- https://github.com/qizekun/PointVisualizaiton
- https://github.com/huggingface/transformers/
- https://github.com/pytorch/pytorch

References

[1] P. Xu, X. Zhu, and D. A. Clifton, "Multimodal learning with transformers: A survey," *IEEE Transactions on Pattern Analysis amp; Machine Intelligence*, vol. 45, no. 10, pp. 12 113–12 132, oct 2023.
[2] OpenAI, "Gpt-4 technical report," 2023.
[3] "Gemini - google deepmind," Dec 2023. [Online]. Available: https://deepmind.google/technologies/gemini/
[4] L. Xue, M. Gao, C. Xing, R. Martín-Martín, J. Wu, C. Xiong, R. Xu, J. C. Niebles, and S. Savarese, "Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 1179–1189.
[5] C. C. Aggarwal, *Neural Networks and Deep Learning*. Cham: Springer, 2018.
[6] J. Sun, Q. Zhang, B. Kailkhura, Z. Yu, C. Xiao, and Z. M. Mao, "Benchmarking robustness of 3d point cloud recognition against common corruptions," *arXiv preprint arXiv:2201.12296*, 2022.
[7] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob, "Mitsuba 2: A retargetable forward and inverse renderer," *ACM Trans. Graph.*, vol. 38, no. 6, nov 2019. [Online]. Available: https://doi.org/10.1145/3355089.3356498
[8] "Pointvisualizaiton." [Online]. Available: https://github.com/qizekun/PointVisualizaiton
[9] W. Wang, V. W. Zheng, H. Yu, and C. Miao, "A survey of zero-shot learning: Settings, methods, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, jan 2019. [Online]. Available: https://doi.org/10.1145/3293318
[10] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.
[11] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[12] J. Merullo, L. Castricato, C. Eickhoff, and E. Pavlick, "Linearly mapping from image to text space," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=8tYRqb05pVn

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[14] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual MLP framework," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=3Pbra-$_u$76D

[15] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 16 259–16 268.

[16] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 19 313–19 322.