Point cloud transformers for 3D fragment matching

Faculty of Information Engineering, Informatics, and Statistics
Master in Data Science

Advisor: Prof. Simone Scardapane

Sottile Alessandro 1873637

Co-Advisor: Alessandro Baiocchi

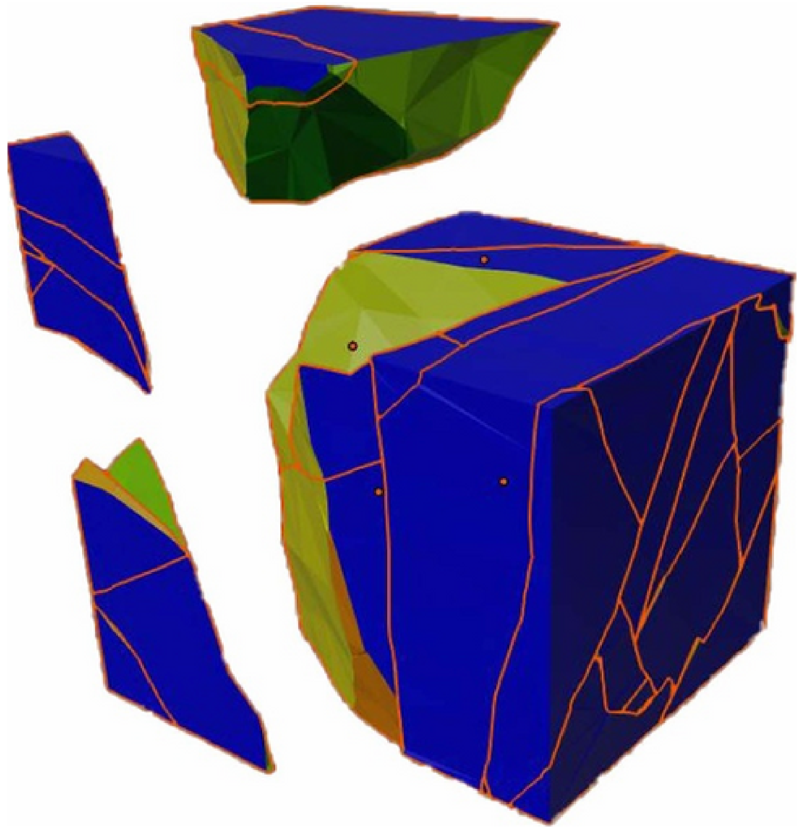
Academic Year 2023/24

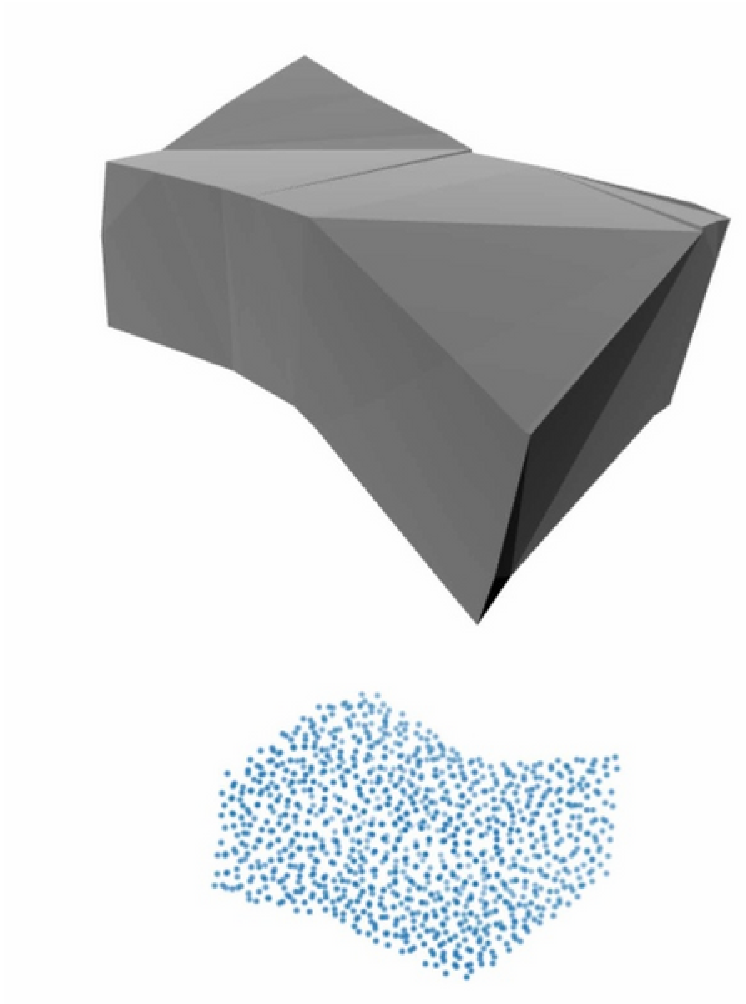
Introduction

- There is a growing trend in scientific papers published annually that simultaneously include tags related to both artificial intelligence and archaeology.
- The goal of this thesis is to build a model capable of performing a novel task: 3D fragment matching. The model is intended to offer support to archaeologists in the job of reconstructing fragmented artifacts.
- This work extends the previous study conducted by Alessandro Baiocchi et al., whose contributions were the creation of a repository of synthetic datasets containing 3D object fragments, called Broken3D, and the execution of the task of internal/external fragment classification.

A. Baiocchi, S. Giagu, C. Napoli, M. Serra, P. Nardelli & M. Valleriani, "Artificial neural networks exploiting point cloud data for fragmented solid objects classification", Machine Learning: Science and Technology, 2023

Dataset





- The data used in this work is the Rotated cuts dataset (RCD), from the Broken3D archive.

https://deeplearninggate.roma1.infn.it/

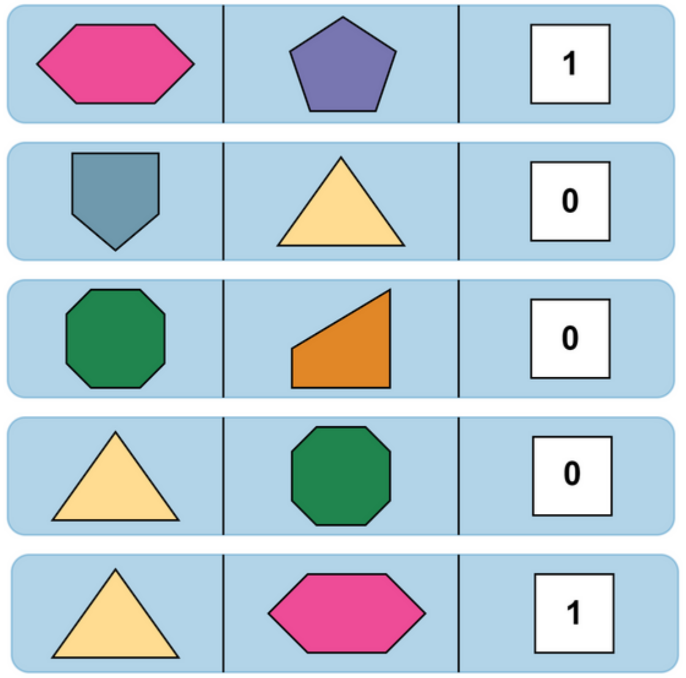
- The dataset is composed of clusters, each having two macro-elements: the fragment collections and the adjacency matrix.

Dataset

- Clusters are unrolled to produce a dataset consisting of triplets [frag_a, frag_b, label].

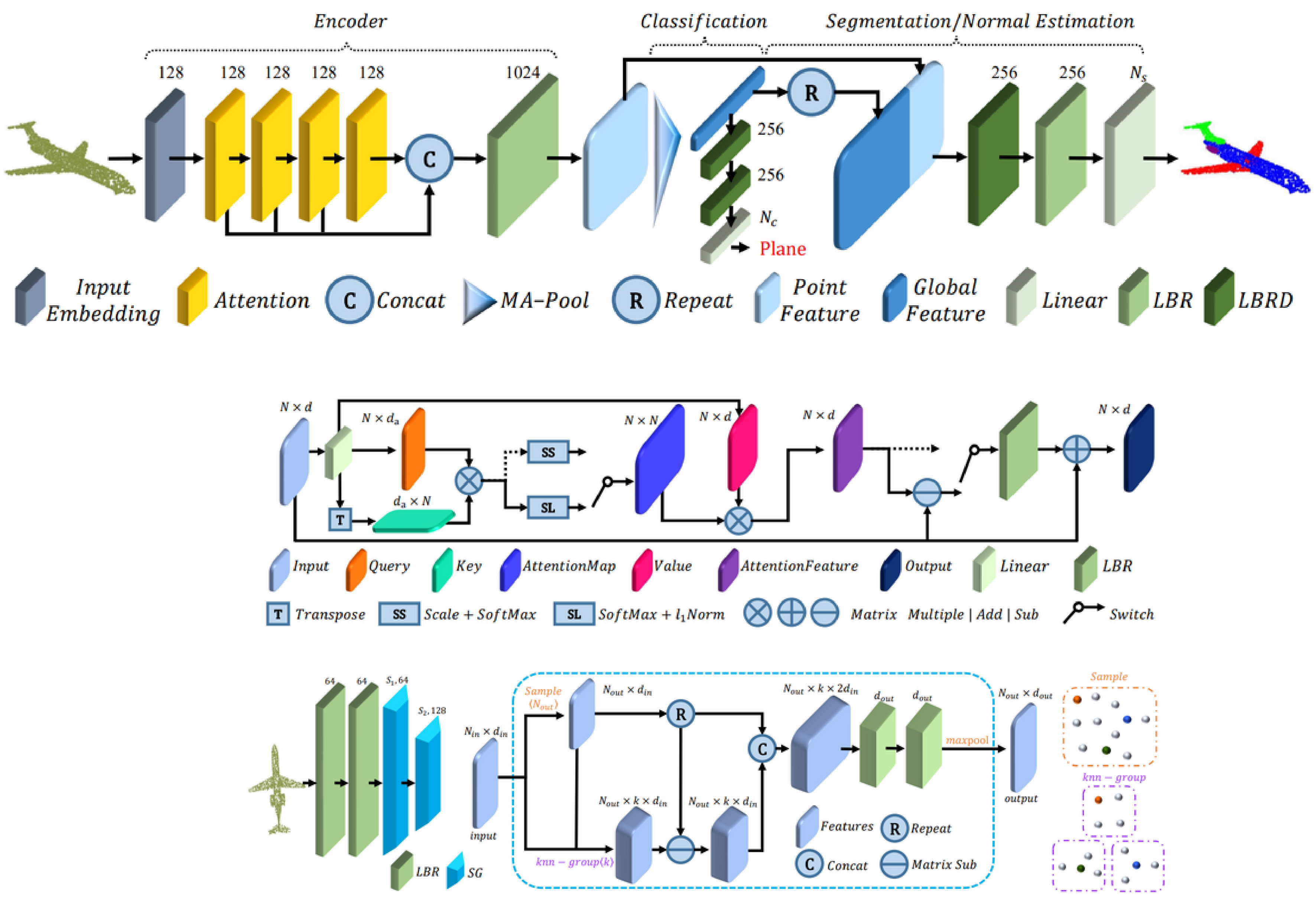
- Each fragment has the shape [1024, 7], the seven features are: [x, y, z, nx, ny, nz, A].

- The dataset is divided Into 70% Train, 15% Validation and 15% Test sets.



Label	Frequency
0	0.857
1	0.143

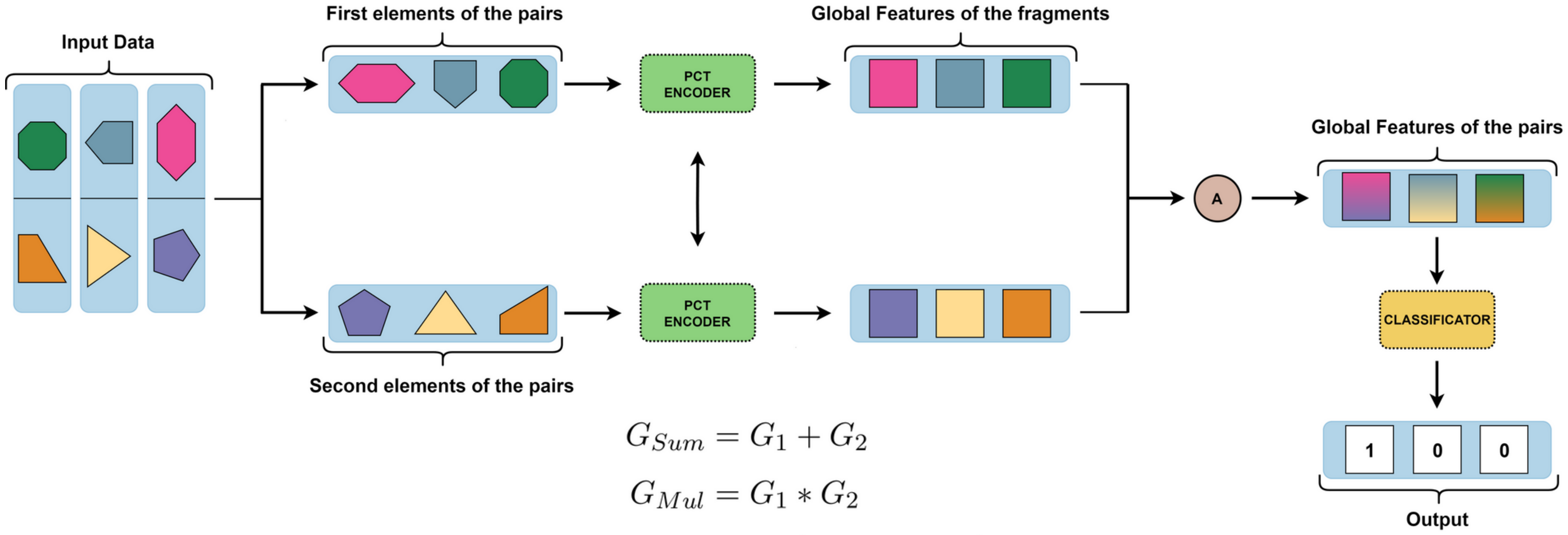
Point Cloud Transformer (PCT)



- The PCT allows the use of Transformers to 3D data.
- The architecture shares the same philosophy as the original Transformer, except for the use of offset attention and Neighbor Embedding in the encoder.
- The offset-attention calculates the differences between the self-attention and input features. Neighbor Embedding is similar to Edge Convolution in DGCNN.

Guo Meng-Hao, Cai Jun-Xiong, Liu Zheng-Ning, Mu Tai-Jiang, Martin Ralph R. & HuShi-Min, “PCT: Point Cloud Transformer”, Computational Visual Media 7, 187–199, 2021.

Model



$$G_{Sum} = G_1 + G_2$$

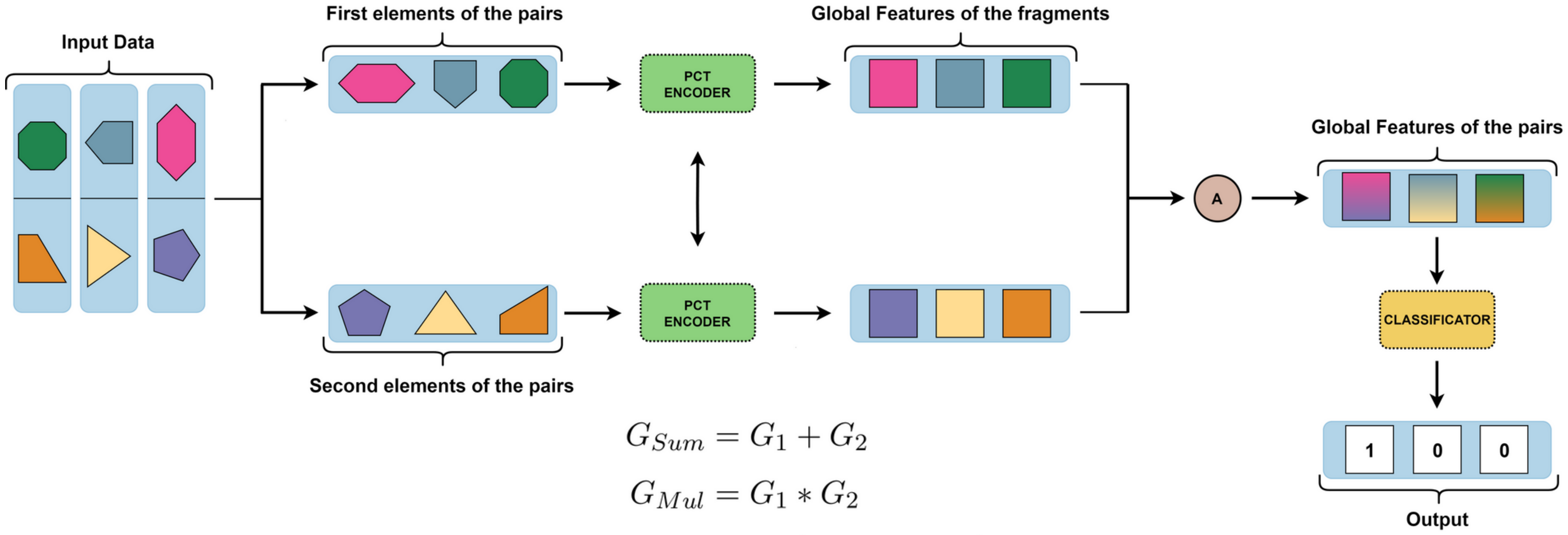
$$G_{Mul} = G_1 * G_2$$

$$G_{Tot} = cat(G_{Sum}, G_{Mul})$$

- The input pairs are split into two tensors containing only the first and second elements, relatively.

- Each fragment undergoes a random rotation, serving as a form of data augmentation, and a translation to the origin.

Model



$$G_{Sum} = G_1 + G_2$$

$$G_{Mul} = G_1 * G_2$$

$$G_{Tot} = cat(G_{Sum}, G_{Mul})$$

- The tensors are processed in parallel in the two branches of the network by the PCT encoder.
- The two tensors of global features are aggregated and subsequently classified by dense layers.

Training details

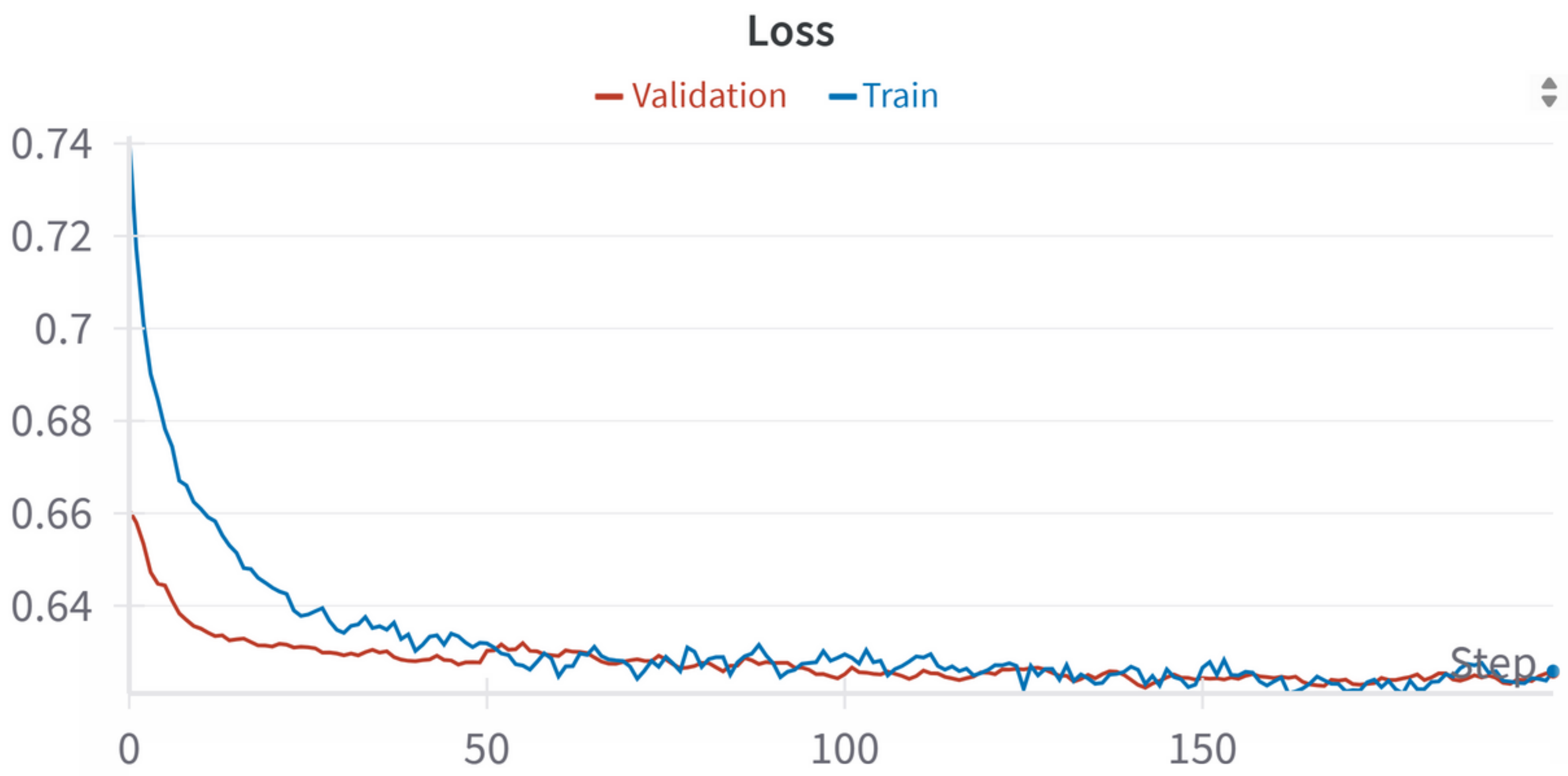
- During training, 10,000 balanced pairs, half adjacent and half not, are sampled at each epoch.
- The validation and test set undergo a subsampling of 6000 balanced couples, which are held fixed in all runs.

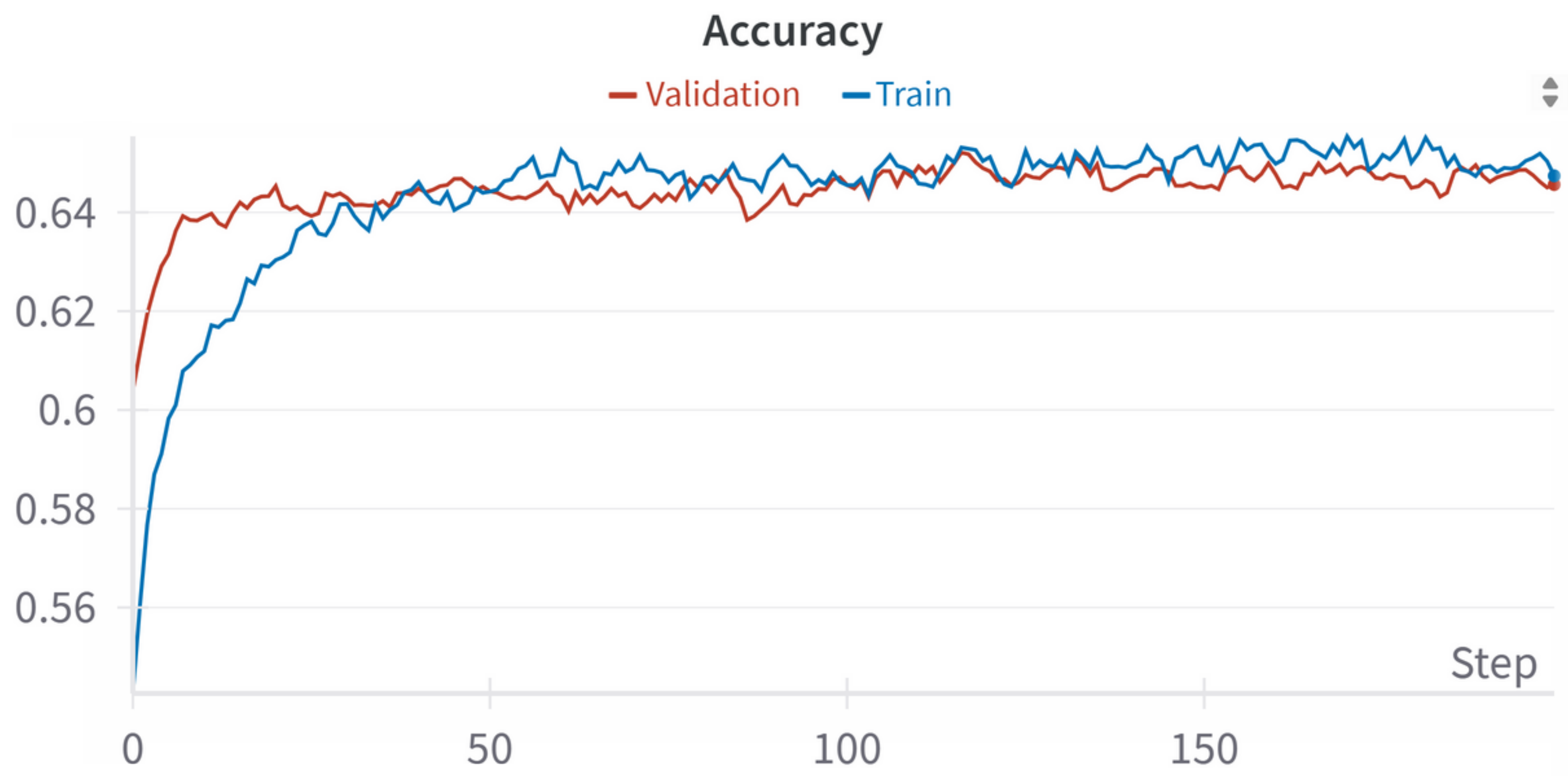
The cross-entropy is employed as the loss function:

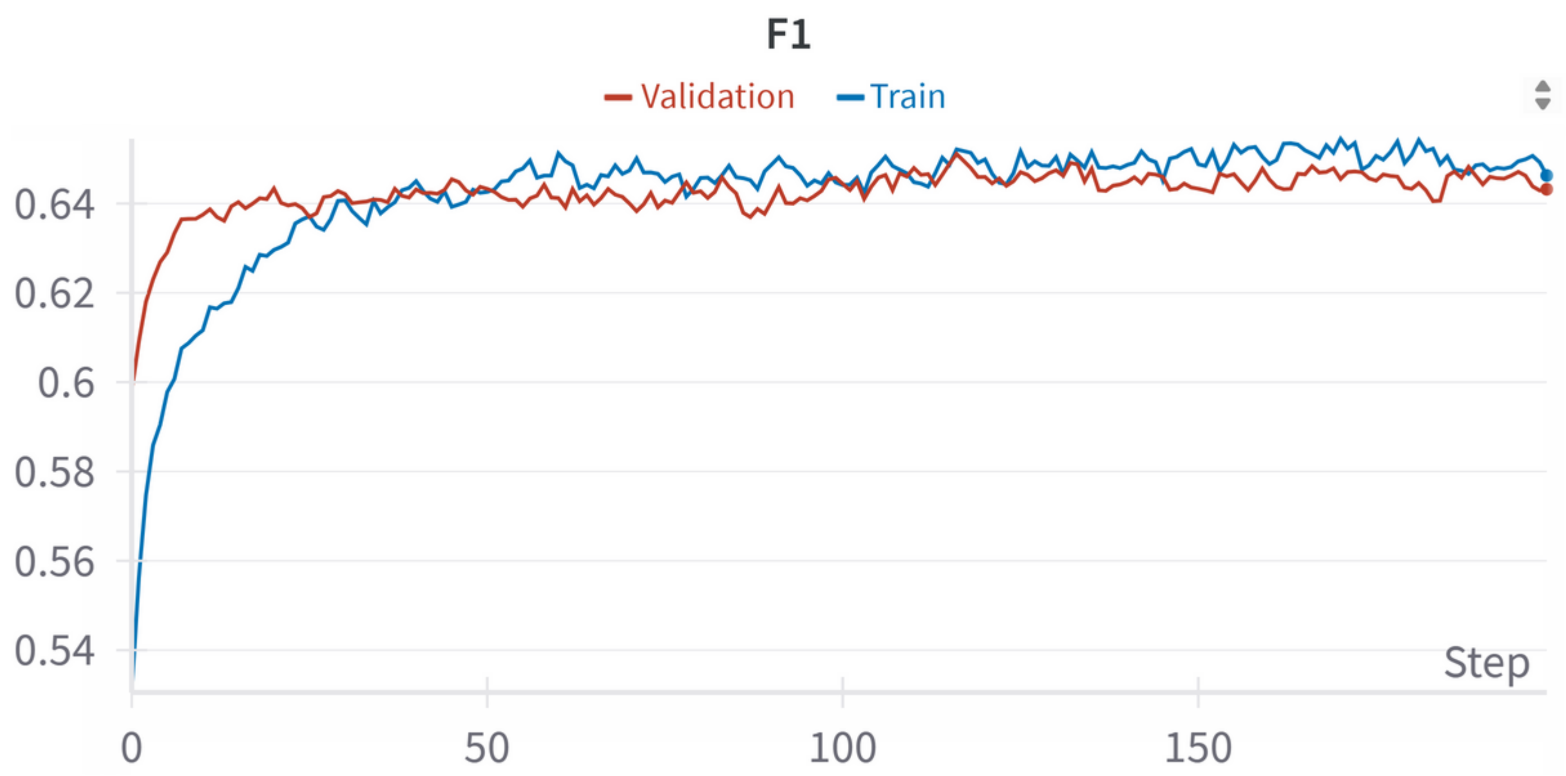
$$Loss = \frac{1}{N}\sum_{i=1}^{n} y_i * log(p(y_i)) + (1 - y_i) * log(1 - p(y_i))$$

Hyperparameter	Value
Batch Size	64
Learning rate	0.00005
Number of Epochs	200
Optimizer	Adam
Weight decay	0.0001
Number of features	7
Number of couples for epoch	10.000
Type of couples	Balanced
Fixed Couples	no

Results: Base Run

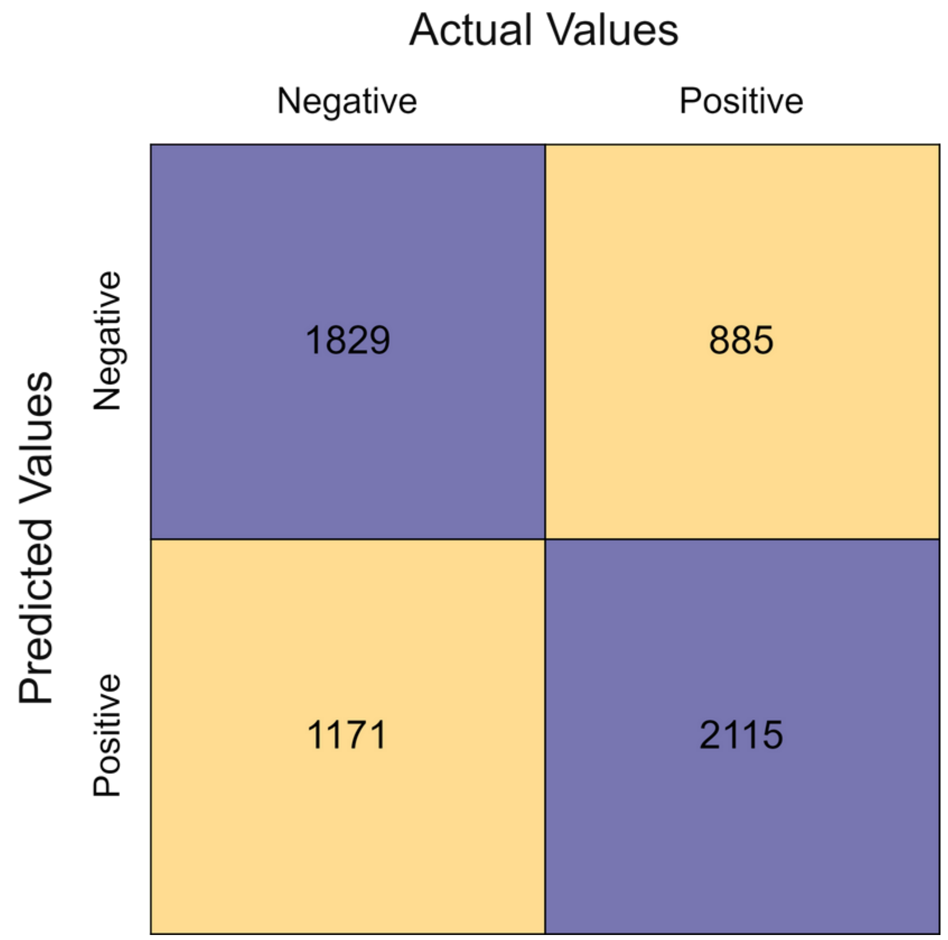






Results: Base Run

- The best results for the validation set are at epoch 116.
- The metrics measured in train, validation, and test sets are very similar to each other.
- The results are good considering also the novelty of the task.



Loss	Accuracy	F1	AUC
0.618	0.657	0.657	0.715

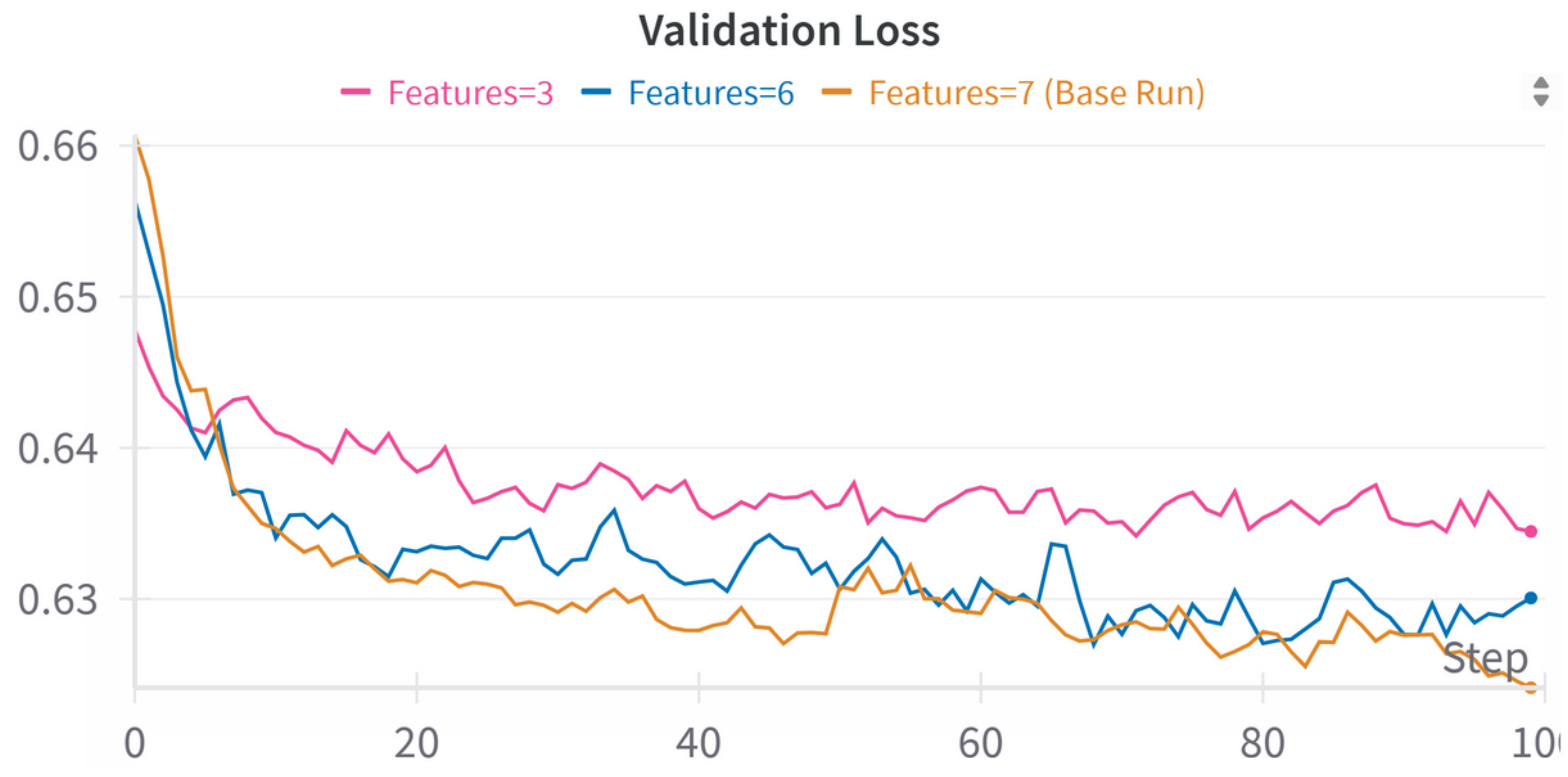
Changing the numbers of features

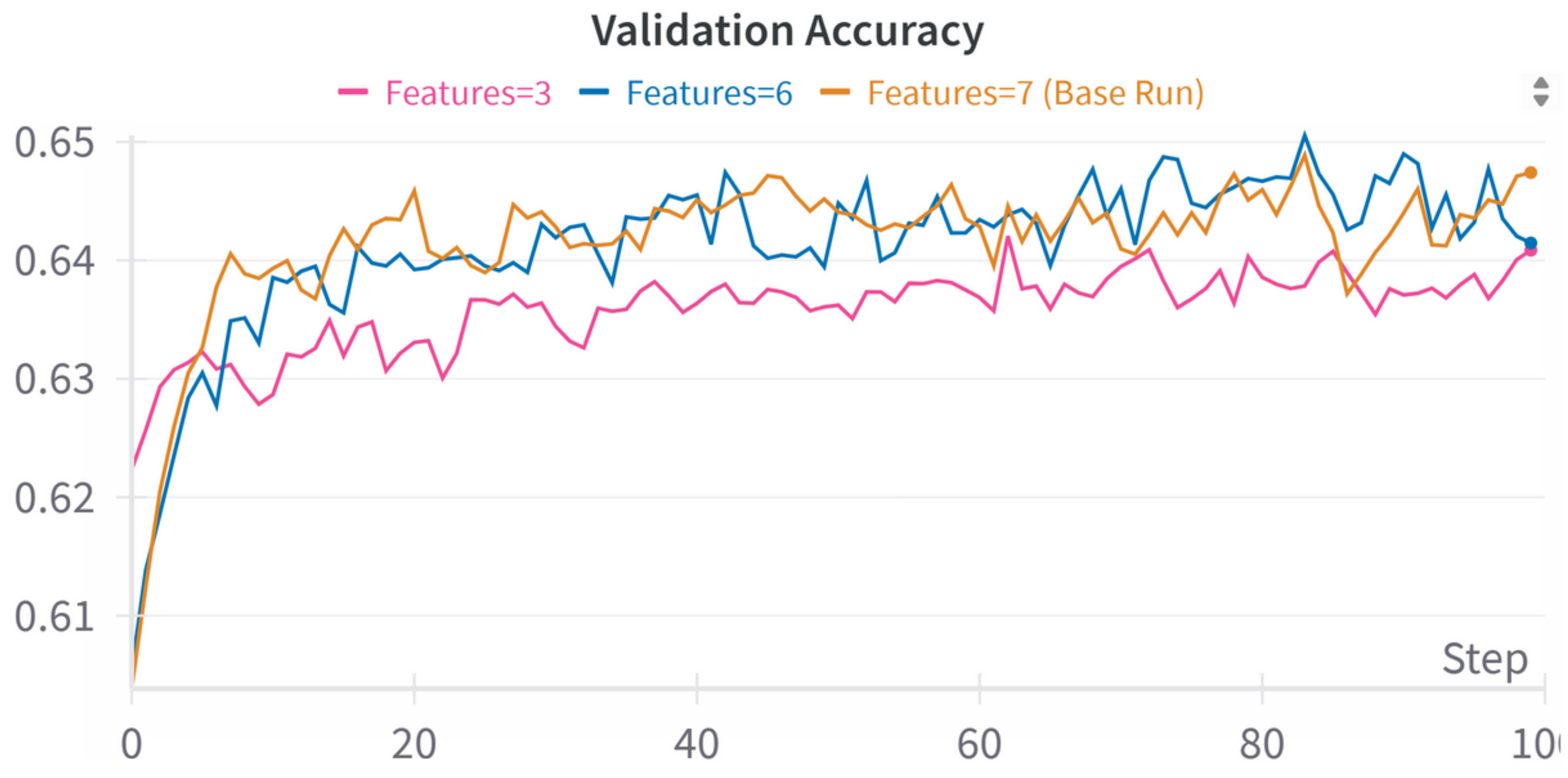
- Two training runs were conducted: one considering only the three spatial features [x, y, z] and the other exclusively removing the triangle area (A) while retaining six features.

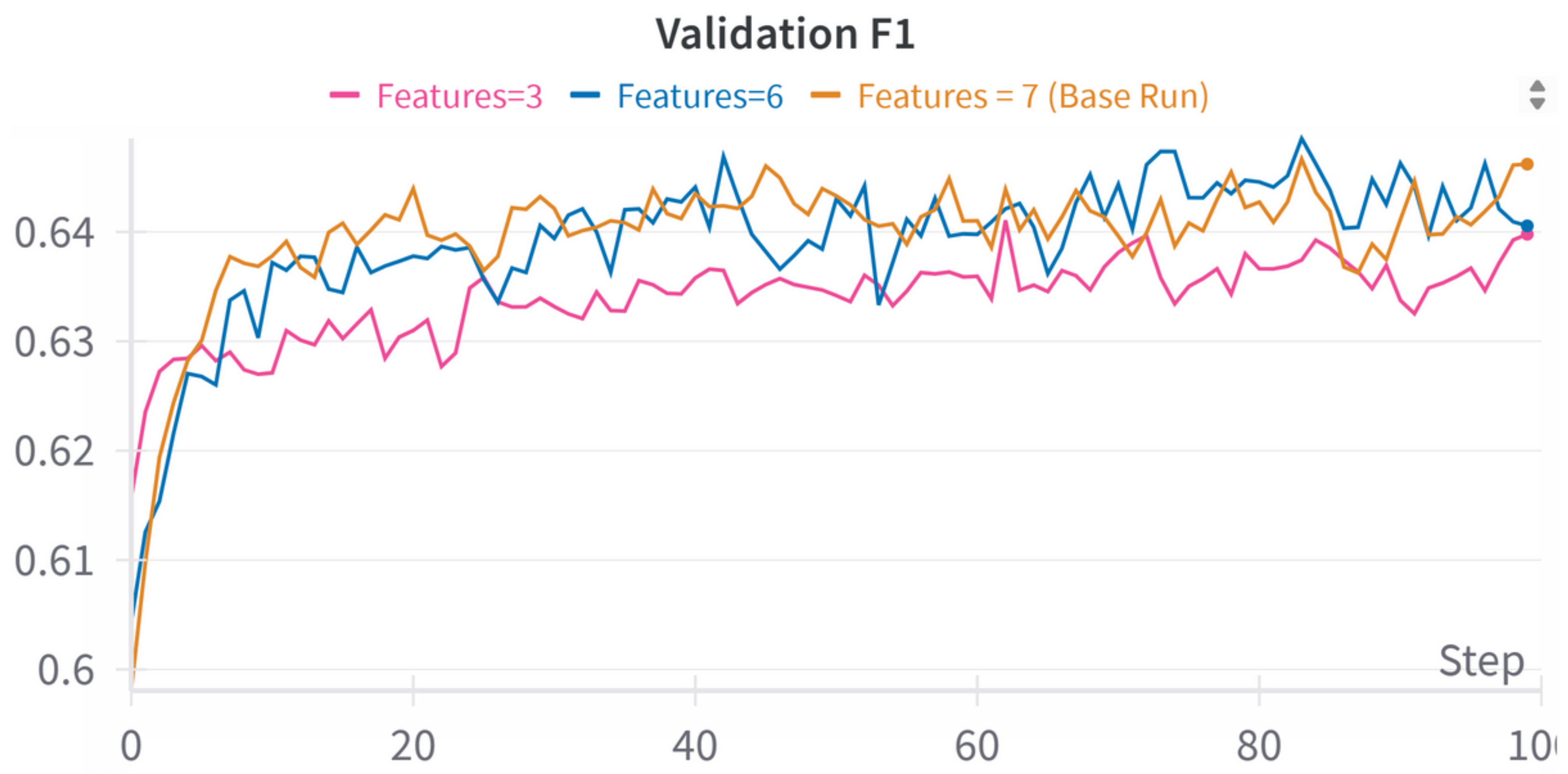
- This analysis aims to provide a more in-depth understanding of the contribution of features within the context of model performance.

- The hyperparameters were kept unchanged from the baseline run, except for the batch size, which was set to 16, and the number of epochs, which was decreased to 100.

Results: Changing the numbers of features



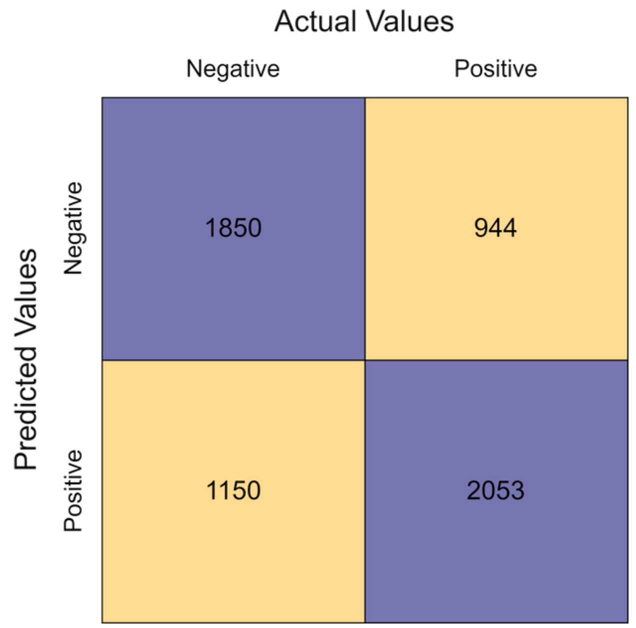




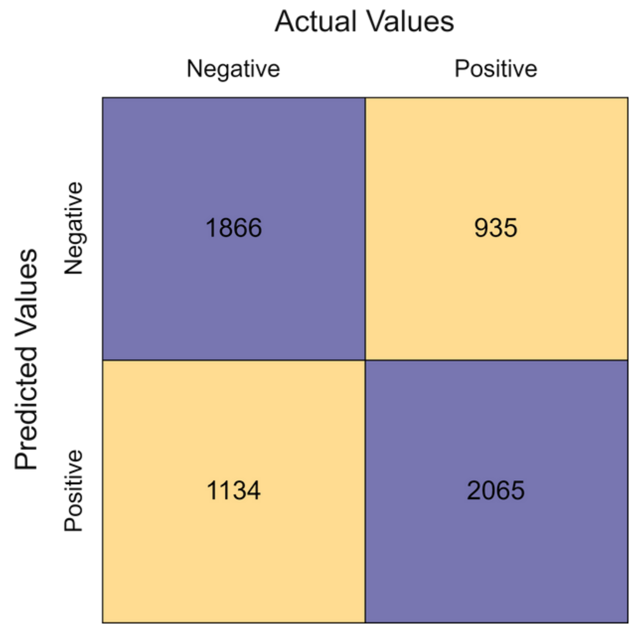
Results: Changing the numbers of features

- The difference between having six and seven features is almost negligible, both in the metrics of the table and when observing the graphs.

- In all three plots, the line representing the scenario with three features consistently produces lower results than the other two. Only at epochs 63 and 100, the values of this run matched the other ones.



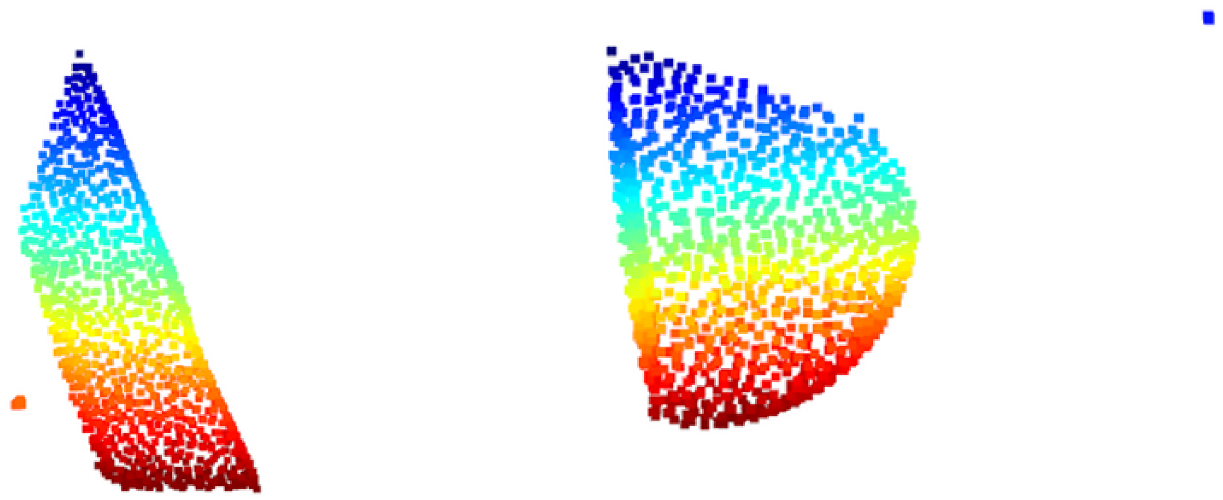
(a) Features = 3



(b) Features = 6

Number of Features	Loss	Accuracy	F1	AUC
3	0.628	0.650	0.649	0.698
6	0.621	0.655	0.655	0.709
7 (Base Run)	0.618	0.657	0.657	0.715

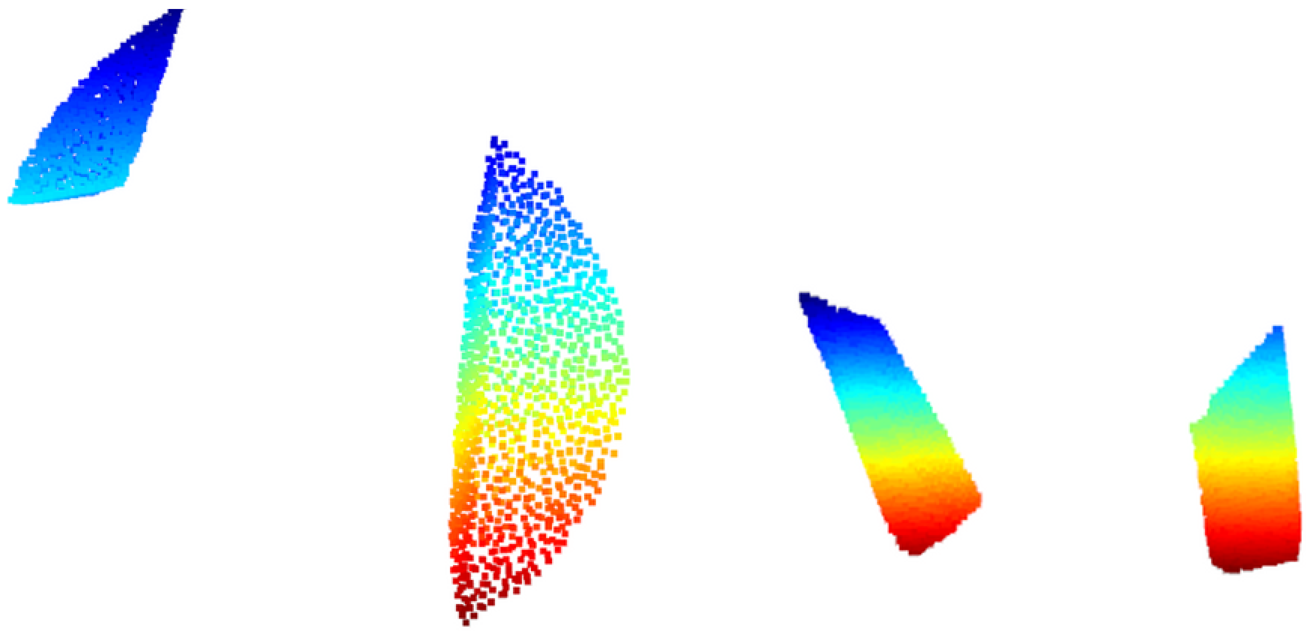
Considerations on model predictions



(a) Couple 5 (b) Couple 1008

- In 10% of the pairs, with rough inference, one of the fragments is significantly larger than the other and the associated label is often 0 (nonadjacent). The model almost always predicts this type well.

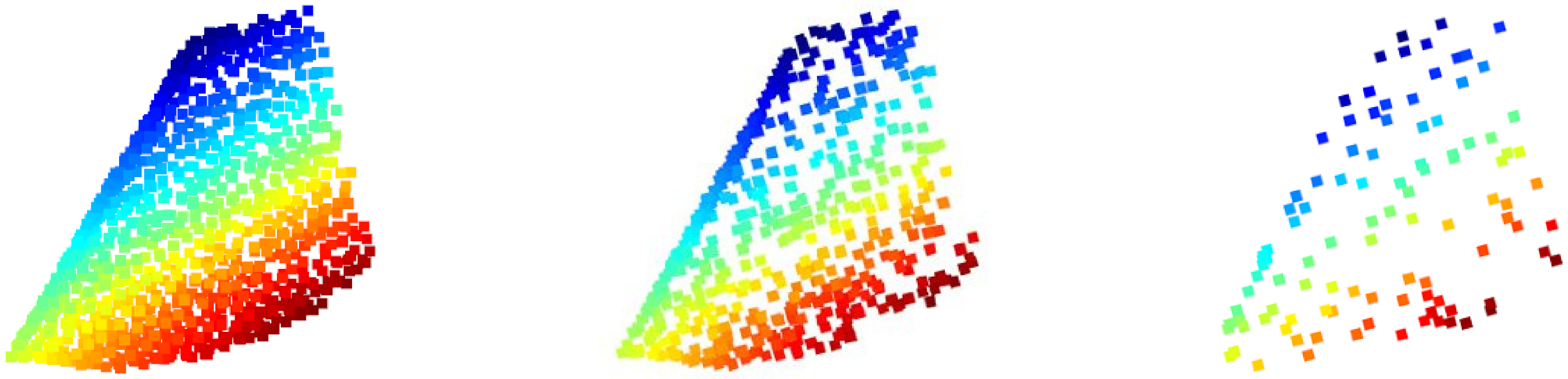
Considerations on model predictions



(c) Couple 1071 (d) Couple 5426

- The network seems to rely on the similarity of the shapes and sizes of the two fragments to make its predictions: the more similar the two objects are, the greater the probability that the model predicts them as adjacent. This approach could be analogous to a puzzle-solving strategy.

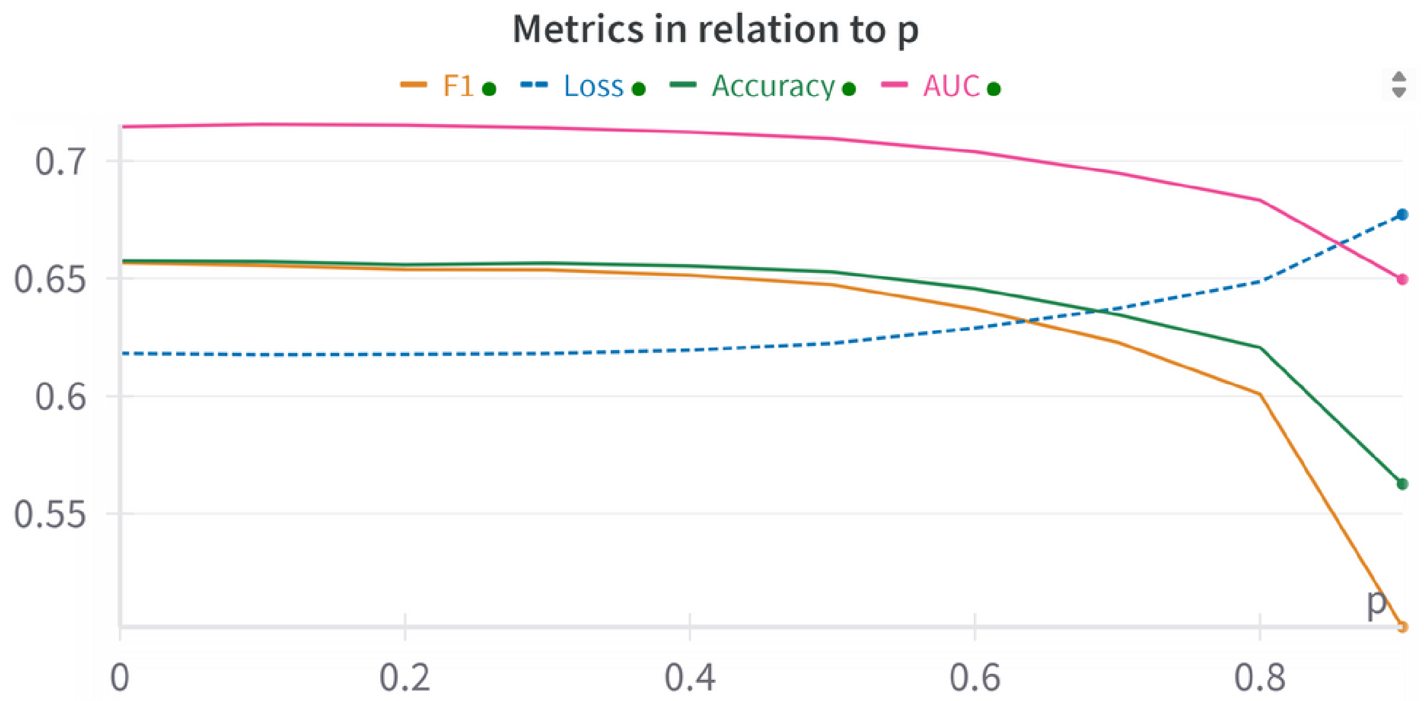
Robustness Analysis: Substituting random points

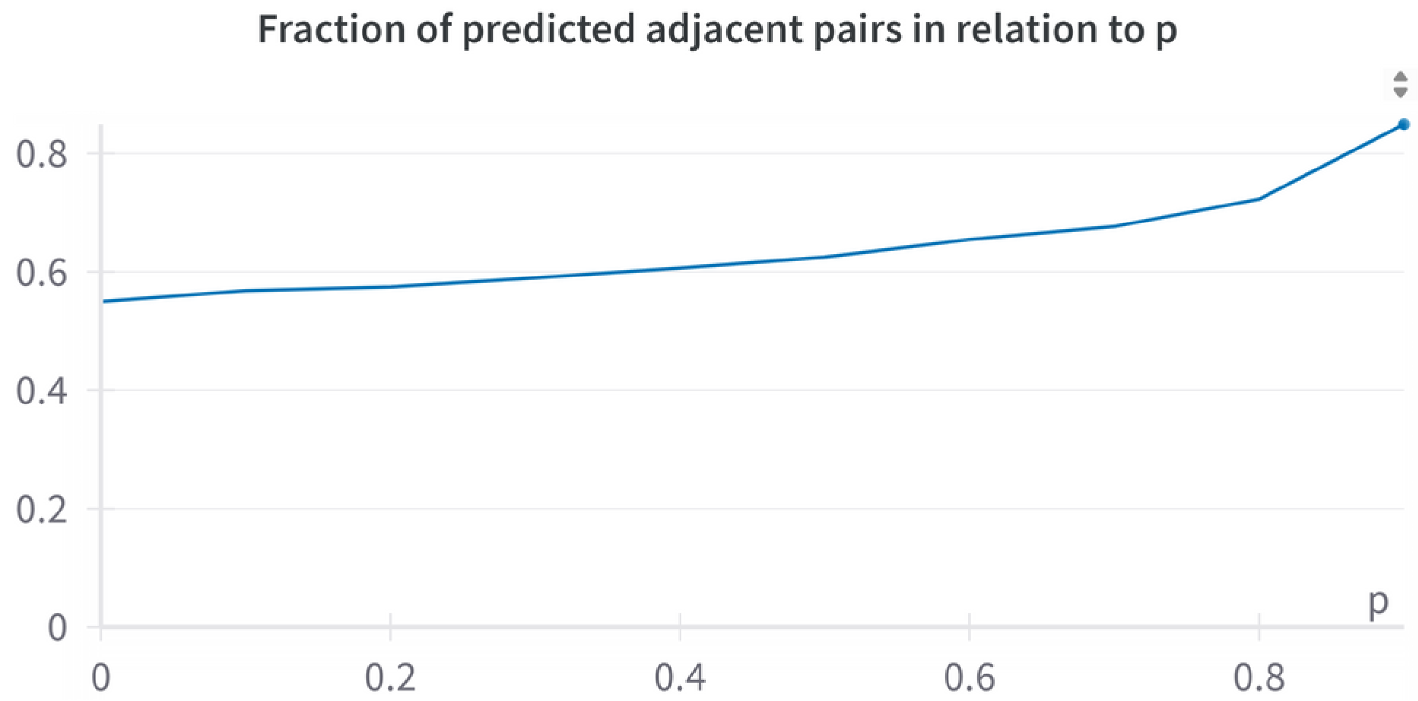


(a) Original Fragment (b) p = 0.5 (c) p = 0.9

- In the analysis, the effect of replacing randomly chosen points with the averages of their respective columns on model performance is evaluated. p represents the percentage of changed points.

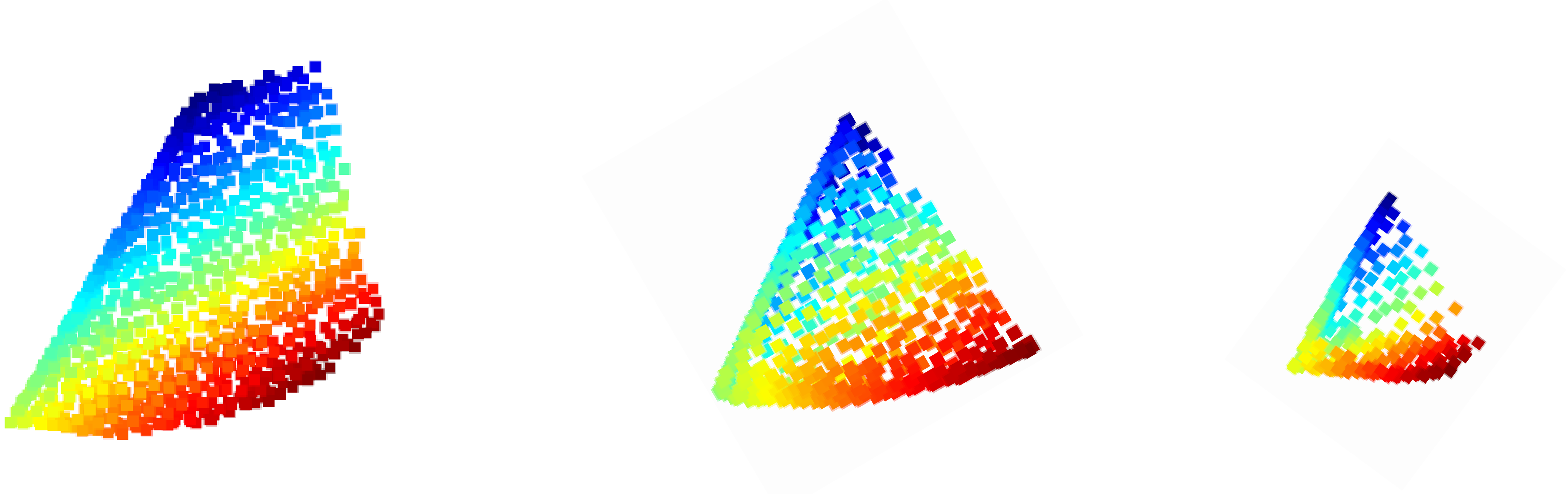
Robustness Analysis: Substituting random points





- Up to half of changed points, the point cloud still retains a lot of detail about the shape of the object represented, and all the performances are more or less the same. Subsequently, there is a drop in model performance and the network begins to predict more and more pairs as adjacent.

Robustness Analysis: Substituting selected points



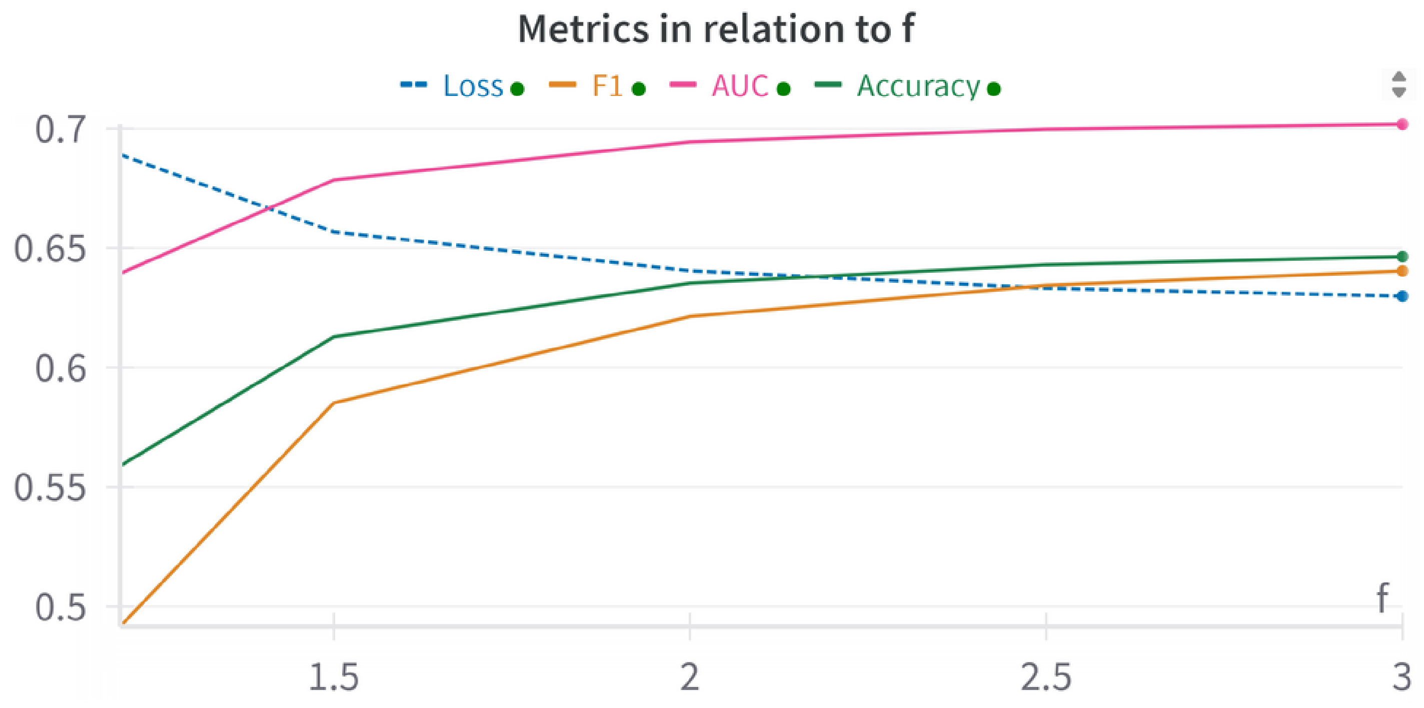
(a) Original Fragment (b) $\theta = 1, \; f = 3$ (c) $\theta = 1, \; f = 1.5$

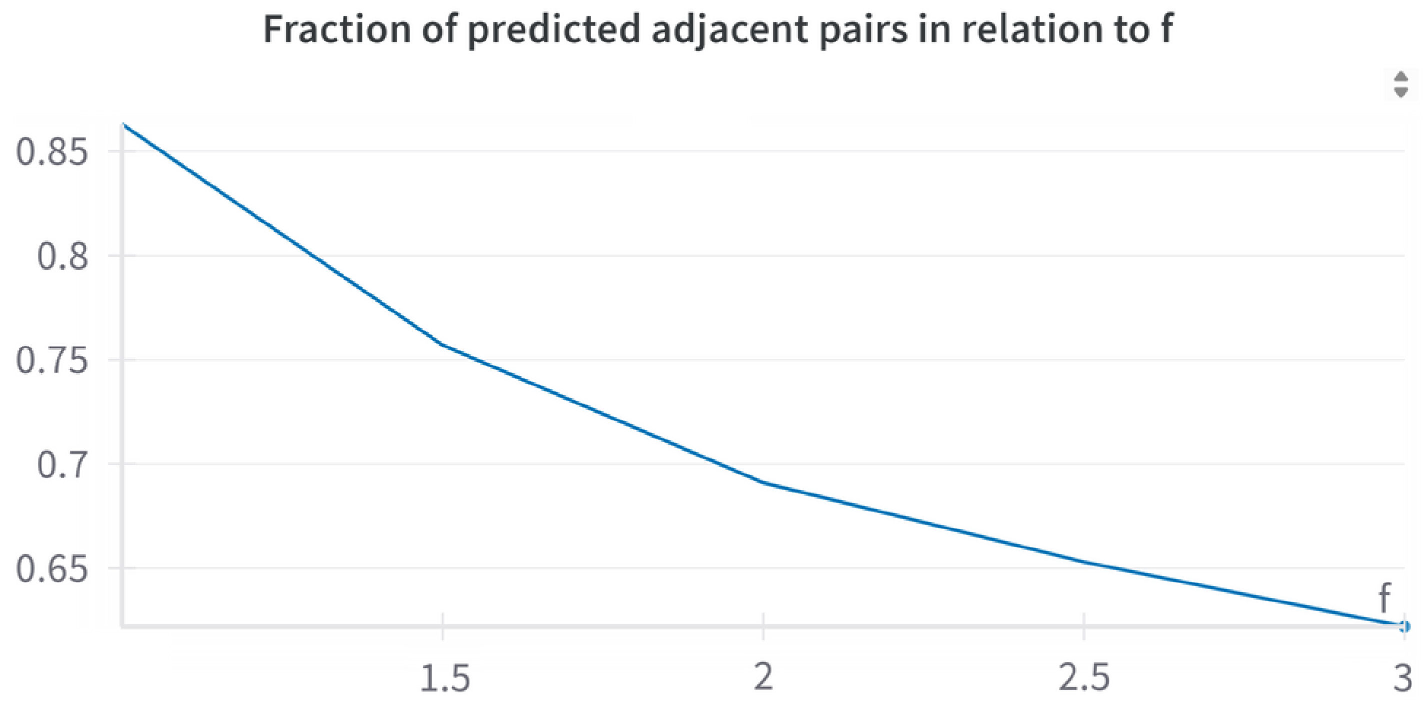
- Entire regions falling within a coordinate interval are obscured. To dynamically select for different fragments the points to modify S_{θ_i} , based on the θ axis, the two extremes are handled as follows:

$$S_{\theta_i} = [min(\theta_i) - 1 < \theta < min(\theta_i) + \epsilon_{\theta_i}]$$

$$\epsilon_i = \frac{max(\theta_i) - min(\theta_i)}{f}$$

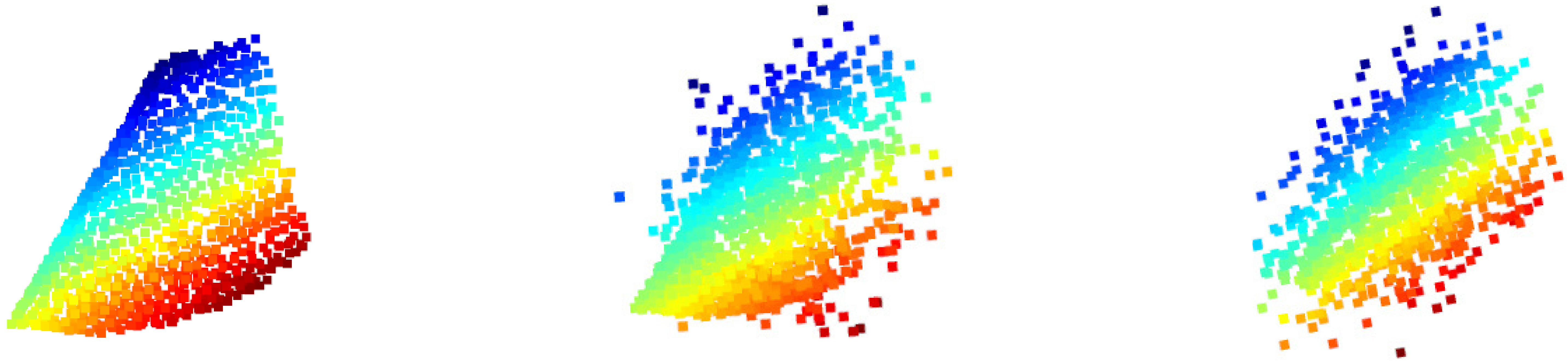
Robustness Analysis: Substituting selected points





- As the number of modified points increases, performance gets worse. The decline is more rapid than before.
- Considering a generic pair, changing the obscured surface could also vary the model's prediction.

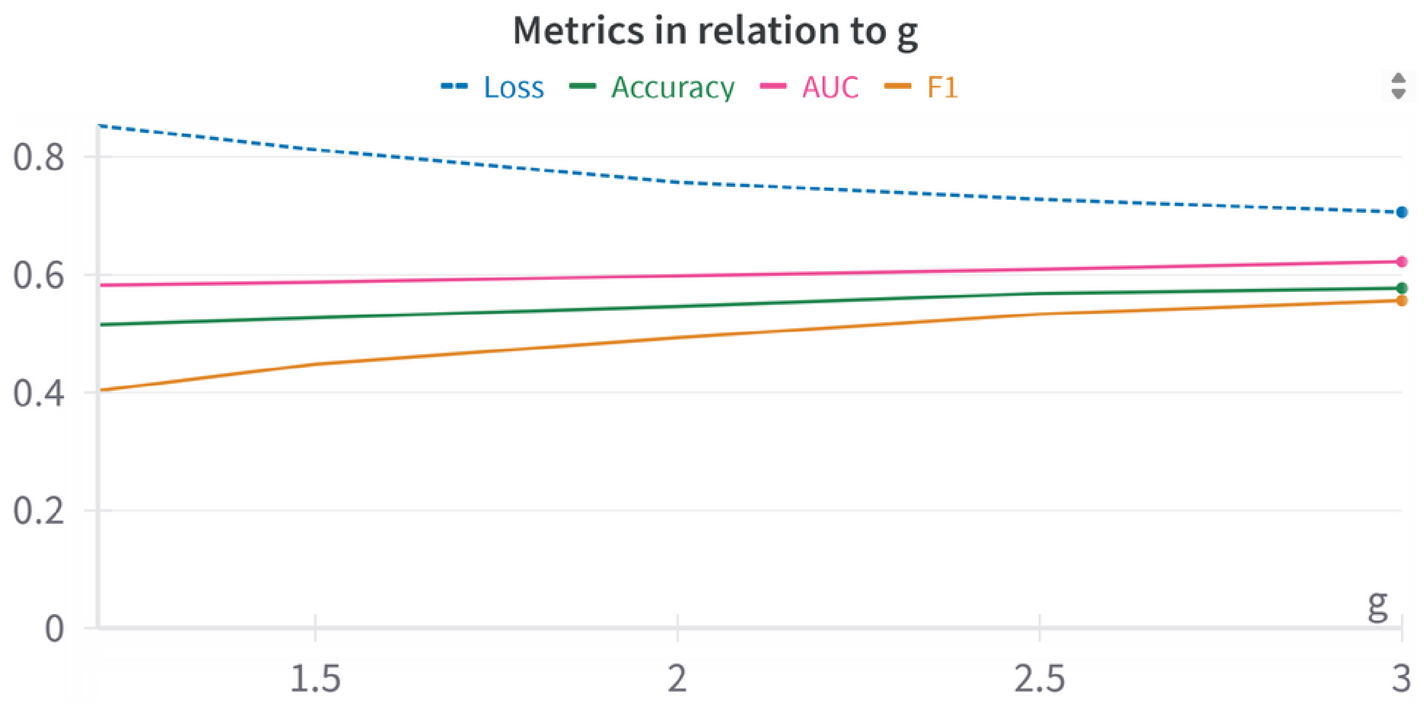
Robustness Analysis: Substituting selected points with generated ones

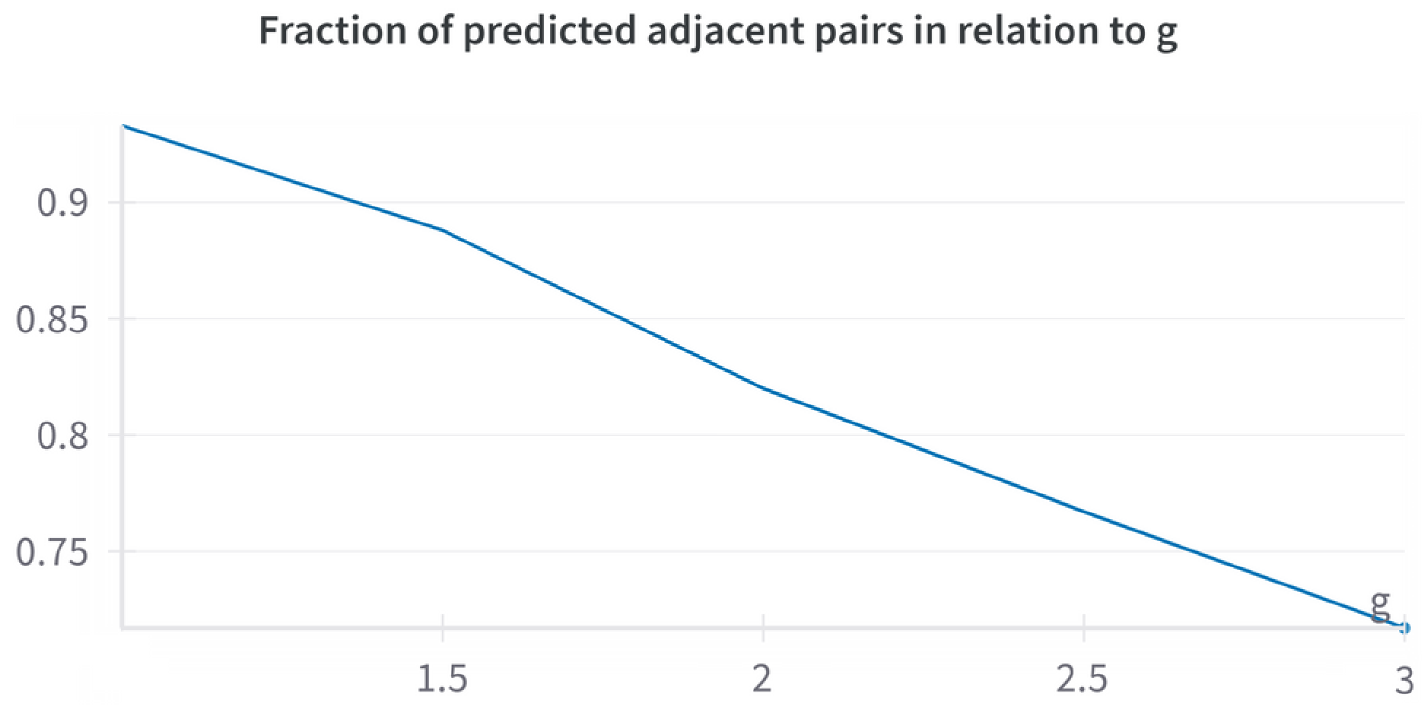


(a) Original Fragment (b) $\theta = 0$, $g = 3$ (c) $\theta = 0$, $g = 1.2$

- Coordinates of selected points are modified by adding noise, thus simulating the generation of new ones. The procedure for selecting points remains the same as before. f is now called g.

Robustness Analysis: Substituting selected points with generated ones





- The metrics immediately assume extremely low values, even with only 10% of points changed. The only pairs that the network continues to predict as nonadjacent are almost all of the peculiar type where one of the fragments is extremely different from the other.

Conclusions

- This work introduced a novel method based solely on the use of Transformer-type neural networks. The performances of the model are discrete, also given the novelty of the task.

- The positive impact on the metrics of the three normals has been established, while the triangle areas do not contribute to increased performance. By excluding feature A from the data, it is possible to optimize computing resources and memory use without sacrificing performance.

https://github.com/Sottix99/Point-cloud-transformers-for-3D-fragment-matching

Conclusions

- By graphically analyzing the point clouds, also under the lens of the robustness analysis performed, it can be hypothesized that the network tends to rely on the similarity of shape and proportion among the fragments to make its decisions. Some peculiar pairs were also found, which may have distorted the workings of the model.

- The editing operations performed on the fragments by obscuring portions of them can prove to be a valuable tool in identifying and removing the various anomalous pairs in the dataset.

https://github.com/Sottix99/Point-cloud-transformers-for-3D-fragment-matching

Thank you for your attention!