Reaction report for “POCO: Point Convolution for Surface Reconstruction”
Valeriy Soltan

*What I like about this paper:*

The paper presents the framework and architecture for the next state of the art approach in object and scene reconstruction. It builds upon previous work by proposing a novel method of computing latent vectors at every input point in a point cloud, thereby avoiding uniform, nondescript vectors and instead capturing information closer to the reconstructed surface. I liked how the paper emphasized the strength of this approach, underscoring its ability to excel in density-varied scenarios such as occluded portions of a scene. It also mentions that the model is capable of producing much thinner surfaces as it isn’t prone to filling volumes. Another thing I liked is that the paper provided a substantial amount of background, giving the reader an insight into alternative methods; as well as the learnings and inspirations that were applied in the development of this approach. Furthermore, I really appreciated that the authors opted to reuse an existing architecture for feature computation, instead of developing a new architecture. I think that papers, particularly in the machine learning domain, should be evaluated on the simplicity as well as the performance of the described approach. I recall my neural networks professor mentioning that a lot of publications in this discipline attempt to reinvent the wheel. They develop a new architecture, inspired by certain principles rooted in very high-level or inaccessible theory only to see a marginal improvement in performance that was mostly caused by overfitting or a slightly tuned hyper parameter.

*What I don’t like about this paper:*

I don’t understand why the paper made surface normals optional and didn’t explore that as a possible improvement to model performance. I feel like the cross-entropy loss could be extended from just penalizing incorrect occupancy predictions to also penalizing misaligned surface normals. Another thing that I did not understand is that the paper acknowledges that voxel-based approaches incur a cubic increase in memory requirements; however, I didn’t see them acknowledge the memory impact of storing a latent vector for every input point in a very dense point cloud. Perhaps I am misunderstanding something but this seemed like a glaring oversight, considering that they even have a section titled “Adapting to large size” where there’s mention of GPU memory limits and how that influenced choices for certain model parameters. I also feel like the paper did a poor job explaining why they chose to calculate the relative occupancy using weighted interpolation. They mention that this offers better generalization but I didn’t really understand the intuition behind this design choice.

*Future directions:*

By their own admission, the architecture’s efficacy drops off when the input point cloud has a large variance in point density such as when a part of the scan is missing. While the model exceeds at producing an object with fine detail, it has relatively worse performance in terms of producing a smooth surface. This could be one avenue for future improvement. I also wonder if there could be a way to dynamically sample a query neighborhood. For example, suppose you have a part of the scan missing. If the query point is near that missing chunk, does it make sense to still sample 64 nearest points, even when most of those points will not be involved in that local region of the reconstructed surface? I understand that this does encode some global context in the latent vector but I wonder if this approach is hindering performance.