

Live Deep Learning Feature Recognition for CAD Human-Machine Collaboration

Gerico Vidanes*, David J. Toal†, and Andy Keane‡
University of Southampton, Southampton, United Kingdom

Xu Zhang§
Falmouth University, Cornwall, United Kingdom

Jon Gregory and Marco Nunez
Rolls-Royce plc, Bristol, United Kingdom

Deep learning models are traditionally trained on large labelled datasets and are static during deployment. Even in standard applications, this is not ideal as input distribution shifts occur - new inputs are slightly different to training samples. This work tackles automated feature recognition in the engineering domain. Here, not only does the input distribution shift but the output space can change as new designs with new geometric features are produced, or the same geometry is viewed in a different context. To tackle this with deep learning, a human-in-the-loop approach is employed to provide an incremental and weak supervision signal to be used by the model to quickly learn a new output space. The continuously learning model is then used to provide suggestions for further features to recognise, effectively multiplying the user effort and accelerating the manual tagging task. The main contribution of this work is a proof-of-concept which suggests that a sparse ground truth signal and limited training can produce a model which provides *useful* predictions. Stochastic simulations are used to model user interaction and empirical results are presented which illustrate the system's learning progression. Results suggest that a model (even from random initialisation) which continuously learns from user inputs can approach the performance of a model statically trained with full supervision on the target task.

I. Nomenclature

$\mathcal{L}_{total}, \mathcal{L}_{point}, \mathcal{L}_{face}, \mathcal{L}_{sparse}$	= scalar neural network losses
\mathcal{P}	= set of points sampled from a geometry
\mathcal{F}	= set of boundary representation faces in geometry
\mathcal{P}_s	= sparse subset of points sampled from a geometry which have been given labels
\mathcal{F}_s	= sparse subset of boundary representation faces in geometry which have been given labels
\mathbf{Y}	= ground truth label of entity
$\hat{\mathbf{Y}}$	= predicted label of entity

II. Introduction

A. Context

Recent applications of deep learning have shown that learning from user data and replicating tasks results in a multiplication of productivity as the knowledge worker and the machine collaborate [1]. Offloading tedious work from engineers frees up time for more creative or more valuable work in general and accelerates the design process as a

*PhD Candidate, Computational Engineering and Design Group, g.vidanes@soton.ac.uk.

†Associate Professor, Computational Engineering and Design Group, d.j.j.toal@soton.ac.uk.

‡Professor, Computational Engineering and Design Group, andy.keane@soton.ac.uk.

§Senior Lecturer, Games Academy, daniel.zhang@falmouth.ac.uk.

whole. The major computer-aided engineering (CAE) software companies are particularly aware of the possibilities as they try and make their products smarter with modern machine learning [2–4].

Geometric feature recognition is the process of identifying portions of a geometry which match the characteristics of interest for a given application [5]. This has become an increasingly important capability in the digital engineering age as it allows for downstream automation. Examples include: mesh refinement, boundary condition or post-processing plane specification [6], predictive selection [4], geometry idealisation for simulation [7], and automated preparation for machining [8].

This task (referred to as semantic segmentation in the deep learning field) has been recently tackled in the literature with some success using deep learning approaches [8–16]. However, the traditional paradigm which involves training static models with large labelled datasets is not always conducive to computer-aided design (CAD). The engineering domain is constantly evolving - new geometries with new features appear as novel designs are developed. Even without this, the geometric features of interest within the same CAD model can be different across applications and studies [5, 6]. For example, structural simulation and fluid simulation will require qualitatively different features to be identified for boundary condition specification. One might also want to identify geometric features within the same CAD model under other contexts, namely geometry simplification for simulation or identification of machining features for computer-aided manufacturing.

Therefore, for many applications, the deep learning model needs to adapt not only to input domain shifts (where the input at test time is slightly outside the training distribution) but also to changes in the semantic output space (the specific features being recognised changes). Inspired by recent works on weak supervision [17] and incremental learning [18], as well as the development of suggestive selection and autocomplete, this work explores the possibility of using a deep neural network (NN) to learn from user inputs in real time and provide suggestions to accelerate feature recognition on previously unseen data.

B. Scenario

Manual feature recognition within CAD most often involves tagging boundary representation (b-rep) faces with meaningful names in a graphical user interface (GUI). The current work explores enabling technologies for the scenario where the user and the learning system collaborates to speed up this manual tagging. Specifically, the live user inputs are used as a sparse training signal to adapt the NN weights incrementally to the domain shift. In the process, the NN predictions are used to suggest labels for as yet unlabelled faces. It is easy to see that a fast learning and accurate model would accelerate the process by multiplying user effort.

The assumptions of this work in terms of a use case are as follows. The user requires a limited number of geometries to be labelled, where curating a large labelled dataset for training a NN to automate this is impractical (in other words, the making of this training dataset would complete the task anyway). A common case in the design stage is also one or two very large geometries to be tagged. The labelling task involves a new context - new geometric features, different analysis, etc. There are pretrained models available but not for this specific task (label space) or for these input types.

C. Contributions

This work stands as a proof-of-concept to start the discourse on live learning CAD assistants. Specifically, to the authors' knowledge, this is the first proof that weak supervision and incremental learning could adapt a NN to a new label space (i.e. new semantics/context). Click-based weak supervision is used as the user interaction; in this case, the 'objectness' prior (an open issue in image-based applications) is somewhat alleviated by the b-rep face discretisation of the geometry. This work presents empirical results from Monte Carlo simulations to illustrate the 'helpfulness' of the system given some preliminary assumptions.

III. Related Work

A. Feature recognition in CAD

There is a breadth of algorithmic approaches to feature recognition and Shah et al. [5] provides a full historical review. However, this work focuses on NN based approaches to explore weak supervision and incremental learning. Early NN approaches have been attempted [19, 20] but without recent advances in computation and geometric deep learning these had limited success. The pioneering contemporary work in 3D geometric feature recognition was a voxel convolutional neural network (CNN) approach [10]. This classified the single machining feature present in the

input. Shi et al. [11] and Yao et al. [12] tackle the same task but using multi-view 2D CNNs and point-cloud NNs respectively. These works require a rule-based pre-segmentation step to extract potential individual features from a real part for classification. Other works directly attempt to process an input with multiple geometric features via semantic segmentation. Where a classification is made per entity. Colligan et al. [8], Cao et al. [13], and Jayaraman et al. [16] do this with a graph neural network (GNN) approach - each b-rep face is a node which is given a classification at the output. Lambourne et al. [15] also directly process the b-rep data but use a topological convolution kernel. Alternatively, Vidanes et al. [9] and Zhang et al. [14] use a hybrid point and b-rep face approach; where the geometry is encoded by a point cloud but predictions are made on the b-rep faces. These deep learning approaches show that a NN can learn to recognise geometric features given a sufficiently large labelled dataset. The current work builds on these static models and aims to provide a framework for using them for real-time learning and prediction.

B. Weak supervision and interactive learning

The weak supervision field has emerged to tackle the high burden of labelling datasets for training semantic segmentation models. Most relevant for this work is click-based weak supervision; in the 2D CNN field, this amounts to clicking objects of a given category within an image. This results in a sparse ground truth mask of a few pixels for use with training. Bearman et al. [17], Xu et al. [21] use this technique to train models from scratch and show that they can obtain competitive accuracy even when compared to full supervision with a given annotation budget. A key issue in this space is the ‘objectness’ prior; in other words, stretching a point input to the whole object in the scene. The current work does not face this issue because of its application to b-rep models - a user click provides a point on a b-rep face which in itself is a sufficient discrete entity with some ‘objectness’.

Adjacent to this field is interactive learning, where user inputs are utilised to augment and improve the system’s predictions. An early work by Le Saux and Sanfourche [22] trains classifiers from scratch with online user inputs for environment mapping. This uses hand crafted features, a simple classifier, and a gradient boosting algorithm. The current work can be seen as an extension of this context-dependent classifier idea but for deep NN semantic segmentation. Contemporary works in interactive learning for image segmentation attempt to utilise weak, click-based supervision within interactive learning. Lenczner et al. [23] trains a model to be able to account for user corrections at test time (without retraining model weights). Several works extend this by doing a limited amount of training as each new sparsely labelled input is received [18, 24–27]. This allows the network to adapt to domain shifts or fine-tune to specific classes by interacting with a user. These works have similar aims and general ideas but differ in the details - mainly how they tackle the ‘objectness’ problem and how to regularise the training to avoid overfitting and catastrophic forgetting of the pretraining. The current work aims to apply these same general ideas to the CAD space, with its own intricacies. Additionally of note is the work by Lenczner et al. [28] which uses weakly supervised interactive learning to add a new class to the label space - i.e., learn a new class to segment. Specifically, the output vector is extended and pixels within a certain object that were previously classified as ‘background’ are now given their own label. In contrast to this extension, the current work instead aims to quickly relearn a new output space to adapt to not only new geometric features but also a new context - different semantic labels given the same geometry.

C. User-guided deep learning in CAD

To the authors’ knowledge, work on applying interactive deep learning to the CAD space is sparse. There are those which are similar to semantic segmentation in implementation but not in spirit [29, 30]. These both output suggestions for portions of a CAD assembly and therefore the NN details are equivalent to semantic segmentation prediction, however they are predicting attributes (material and mating type respectively) rather than recognising geometric features. Of interest to the current work, Bian et al. [29] studies how user inputs can be used to inform and guide the prediction of a NN. Similar to the work by Lenczner et al. [23] described above, there is no online retraining involved and the NN is instead trained to be able to receive this extra information as input. Specifically, the user can provide material specifications for a subset of the assembly which is accounted for by the NN when predicting the suggested material for the rest of the assembly. On the other hand, Jones et al. [30] acknowledges that there are many ways to mate parts into an assembly and therefore utilise user input to tackle this ambiguity. Based on the two b-rep faces the user has clicked, the system gives top-6 suggestions for how to mate them. This mechanism was simulated during the NN training. These works account for the user information within the NN input and utilise static pretrained models. The current work seeks to bring the ideas from the image deep learning space to extend human-machine interaction within CAD by learning new contexts through user interaction.

Will there be additional user input during the loop (arrow 1) following the initial user involvement? Or will it proceed to train the neural network directly, taking into account the acceptance/rejection of suggested labels as the user labelling step (arrow 2)?

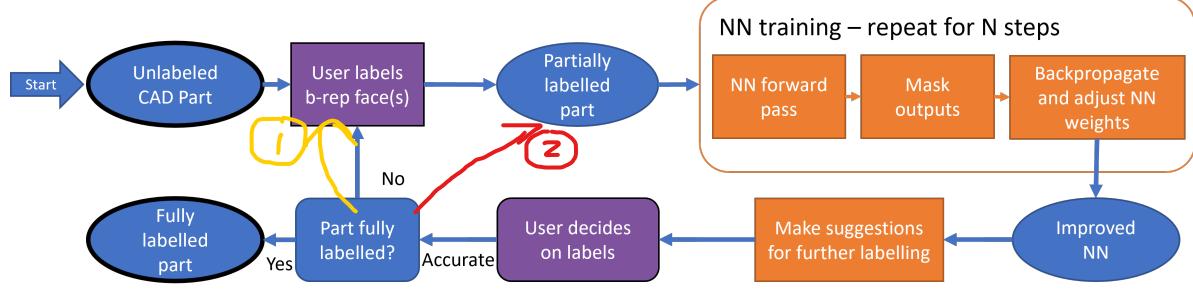


Fig. 1 Proposed proof-of-concept workflow for illustrating a system learning from live user inputs and giving suggestions throughout.

IV. Proposed Approach

A. Deep neural network

The NN model used in this work is based on the *PointNet++* [31] architecture but has been extended to properly process b-rep CAD models, details of this are in the work by Vidanes et al. [9]. The ideas in this paper should be agnostic to the underlying NN model being used, although different models will affect runtime and learning performance. Comparisons of different base NNs is left as future work.

An architectural detail of note here is the loss function. The network described in [9] utilises a branched architecture which jointly predicts semantic labels for each point and each b-rep face. The cross-entropy loss is then averaged across these branches:

$$\begin{aligned} \mathcal{L}_{total} &= \frac{1}{2} (\mathcal{L}_{point} + \mathcal{L}_{face}) \\ &= \frac{1}{2} \left(\frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} (-\mathbf{Y}_p \cdot \log \hat{\mathbf{Y}}_p) + \frac{1}{|\mathcal{F}|} \sum_{f \in \mathcal{F}} (-\mathbf{Y}_f \cdot \log \hat{\mathbf{Y}}_f) \right) \end{aligned} \quad (1)$$

It is speculated that this is particularly conducive to the weak supervision signal (in contrast to purely b-rep NNs) since user clicks correspond to b-rep faces which can then be propagated to multiple associated points. [Proof of this will potentially be explored for the final manuscript].

B. Learning from live user inputs

Figure 1 illustrates the workflow for utilising live user inputs to adapt a NN to a new output space (and potentially input domain). It is assumed that the size of the new output space is given prior to the interaction to initialise the new output vectors. The workflow begins with a user manually labelling a b-rep face within the CAD model; this single partially labelled geometry can then be used to train the NN with a limited number of backpropagation steps. The number of training steps being done can balance the amount of adaptation with runtime [25]. Moreover, in this case, because the network is attempting to learn a new output space with incremental labels the amount of training steps should be limited to avoid overfitting to the sparse labels at the start. With rough guidance from Kontogianni et al. [25], 5 training steps are used here. This should be revisited as future work; however, it was found that an almost arbitrary but constant value is sufficient as a proof-of-concept.

After each round of training, predictions from the updated NN are used to suggest tags for other b-rep faces. These suggestions are then confirmed or rejected by the user. Correct suggestions add to the partial ground truth for later training. At this stage, incorrect suggestions are simply discarded. The workflow repeats as more **user input** is received and ends when the CAD model is fully labelled. *It appears that the user input has been treated as the labelling action, so the loop should avoid returning to the "initial" labelling step.*

For learning from the weak supervision signal, a sparse loss approach is taken similar to [17, 18]. Where the total loss is calculated similarly to equation 1 but only for those entities which have been given labels (through user input or previously correct predictions):

$$\mathcal{L}_{sparse} = -\frac{1}{2} \left(\frac{1}{|\mathcal{P}_s|} \sum_{p \in \mathcal{P}_s} (\mathbf{Y}_p \cdot \log \hat{\mathbf{Y}}_p) + \frac{1}{|\mathcal{F}_s|} \sum_{f \in \mathcal{F}_s} (\mathbf{Y}_f \cdot \log \hat{\mathbf{Y}}_f) \right) \quad (2)$$

At this stage, no regularisation losses are used. The losses to assert the ‘objectness’ prior present in the weak supervision works for images is not needed here because of the ‘discreteness’ of the b-rep faces. Regularisation losses to control the amount of adaptation to new input, used in [18] for example, is also not used here. Intuitively, the training should be unhampered to learn the new output space as fast as possible (downstream prediction layers have been re-initialised to random values).

C. Simulating user interaction and evaluation framework

Following the lead of many works in the interactive learning literature, user inputs are simulated to evaluate the system instead of running large scale user studies. In this work, fully labelled datasets of geometries are used together with stochasticity to simulate perfect but incoherent manual user tagging. It is recognised that **real manual user tagging is more coherent - possibly being localised or focusing on individual classes in sequence** - but more realistic modelling of user behaviour is left as future work. A simple, preliminary model for user behaviour is used here: a random, as yet unlabelled b-rep face is chosen and its ground truth label is revealed and added to the CAD model.

Presenting the NN predictions for all unlabelled b-rep faces to the user can be counter-productive, especially for large geometries with small faces and for the start of the interaction where NN accuracy is poor. A reasonable subset should be presented as suggestions. The optimal set of suggestions to present the user is a problem for investigation in itself and is left for future work. Here, a simple initial baseline is used: predictions for five b-rep faces randomly chosen from those currently unlabelled are presented as suggestions.

Finally, the metric for performance in this case is not as simple as the ‘number of user clicks’ nominally used in the image literature. The works on interactive segmentation for images tend to involve the user correcting initial predictions. In contrast, here the user interaction involves adding labels that the NN can learn from as well as confirming label suggestions. The key assumption here is that confirming a suggested label is significantly less effort than manually tagging that b-rep face. It is difficult to assess this and therefore to assess the performance of the proposed system without timed user studies. At this stage, this work focuses on the potential for fast learning within this weakly supervised and incremental configuration. The assumption is that this correlates to acceleration of assisted manual tagging. The proposed approach is compared to baselines to illustrate relative improvement rather than absolute performance. Therefore, a simple metric is used: the number of repetitions of the workflow in figure 1. Which can also be loosely referred to as the number of b-rep faces which were ‘manually’ labelled within the geometry - in contrast to those labelled with NN suggestions. This can then be normalised using the total number of faces in the geometry to get a percentage metric where lower is better (less ‘manual’ effort).

V. Experimental Details

A. Datasets

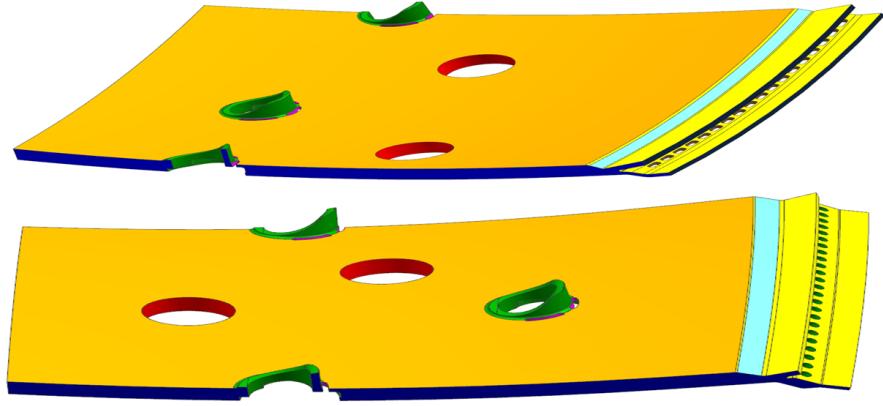
Industry relevant test cases have been used for testing the workflow just described. These are typical parts of gas turbine engines and form realistic test cases where accelerated manual tagging would be useful. The first set of examples are variations of a combustion chamber section, used by [32] in an optimisation study. The parametric CAD model was used to reposition dilution ports on the liner to create 1000 slightly different geometries. **An example is shown in figure 3.**

Not Fig. 3!
should be Fig. 2. A total of 19 classes are used to segment these geometries - an extended version of that used by [32] - and are mostly related to fluid simulation. Examples include the *inner hot skin*, the *outer cold skin*, periodic boundaries, and *chuted ports*. These geometries have around 200 b-rep faces each. The second dataset is over twenty thousand variations of a fuel spray nozzle which was made available by *Rolls-Royce* to the authors as part of an unpublished work*. These have been algorithmically labelled with 9 classes describing high-level geometric features; such as **stem, outer swirler vane, inner swirler vane, and outer swirler case**. These geometries have over 400 b-rep faces each. **refer to Fig. 3 here?**

In addition, the public *MFCAD++* dataset [8] has been used for pretraining the NNs where stated. This is a large (41766 training geometries) and diverse dataset of algorithmically generated shapes containing multiple machining

*As part of the NUSKU project funded by the GEMinIDS programme: GEMinIDS - Geometry Enabled Modelling in Integrated Design Systems - Innovate UK Project 113088.

Very good point!
It's worth investigating in the future.



I couldn't find a reference to Figure 2 in the main text.

Fig. 2 CAD model of sections of the inner and outer liners of an aircraft engine combustion chamber. Different semantic classes are colour coded.

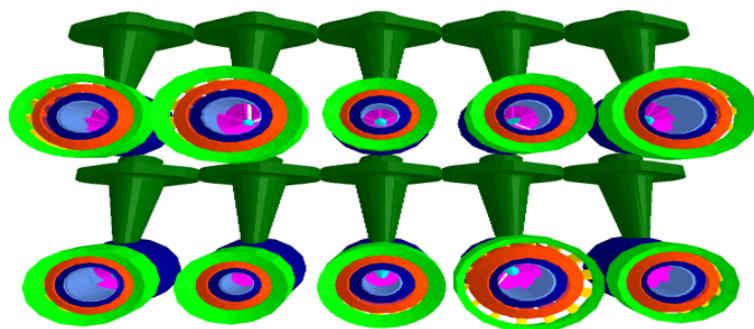


Fig. 3 Examples of fuel spray nozzle geometries as triangular meshes. Originally CAD models but triangulated for simple visualisation. Semantic classes are colour coded.

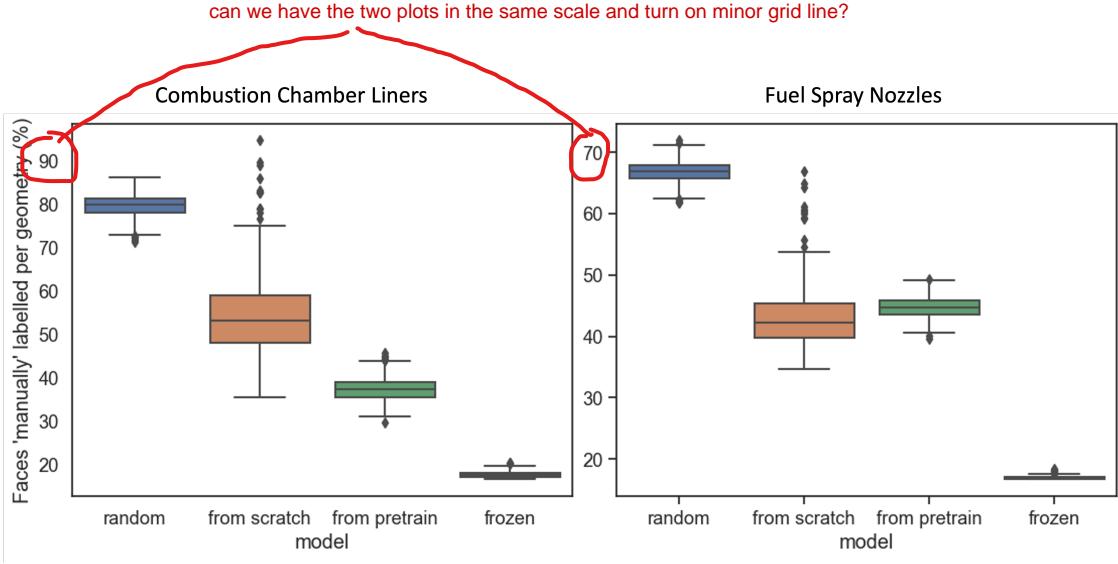


Fig. 4 Boxplots of distributions made by different model configurations when run against the (Left) combustion chamber liners dataset and the (Right) fuel spray nozzles.

features.

B. Baselines

use _ to connect the 2 as 'num_class' for clarification?

Worst case and best case baselines have been used in this work to alleviate the weaknesses of the performance metric and further contextualise the potential of the approach. A random baseline has been used to show the effect, if any, of the user confirmation mechanism on the obtained performance metric. In this case, random integers are sampled from a uniform distribution, $\mathcal{U}[0, \text{num. classes}]$, for each of the five b-rep face suggestions. On the other hand, an ideal case baseline is represented by a model trained with fully supervised learning on a dataset of labelled examples for the appropriate target task. This model is frozen (i.e. no further training occurs during the ‘user interaction’) and is merely used to produce predictions which the five suggestions can be sampled from. It is worth noting that a ‘perfect’ model or oracle which gives fully correct suggestions would always give 16.67% for the current metric - for each added ‘manual’ user label, five NN suggestions are added to the CAD model.

Use Roman numerals 5 to keep it consistent as other places in the paper?

C. Training

Where applicable, pretrained models have been trained with a standard fully supervised approach, described in [9]. The extracted model weights are those which achieved the highest overall b-rep face labelling accuracy across the (in-distribution) validation split of the dataset.

A simple approach to incremental learning has been adopted here where the downstream prediction layers have been re-initialised and all layers continue training with the same configuration as the standard fully supervised approach. Notably, the default learning rate for the ADAM optimiser [33] (1×10^{-3}) is used for all layers. This is likely not optimal but a hyperparameter search is left as future work.

VI. Results & Analysis

A. Learning from single geometries

The experiments in this subsection represents the case where a single (large) geometry needs to be labelled with the interactive learning. To obtain statistical results, the experiments are run across all testing geometries five [more planned for final manuscript] times - to illustrate robustness against the geometry and the randomness from ‘user’ labelling. The learning configurations are reset for each labelling interaction consisting of a single geometry.

Figure 4 shows the performance of each configuration as distributions. Two incremental learning configurations are tested. The first is using a randomly initialised neural network each time and the second is one which was pretrained on the MFCAD++ dataset. Shouldn’t it be Fig. 5?

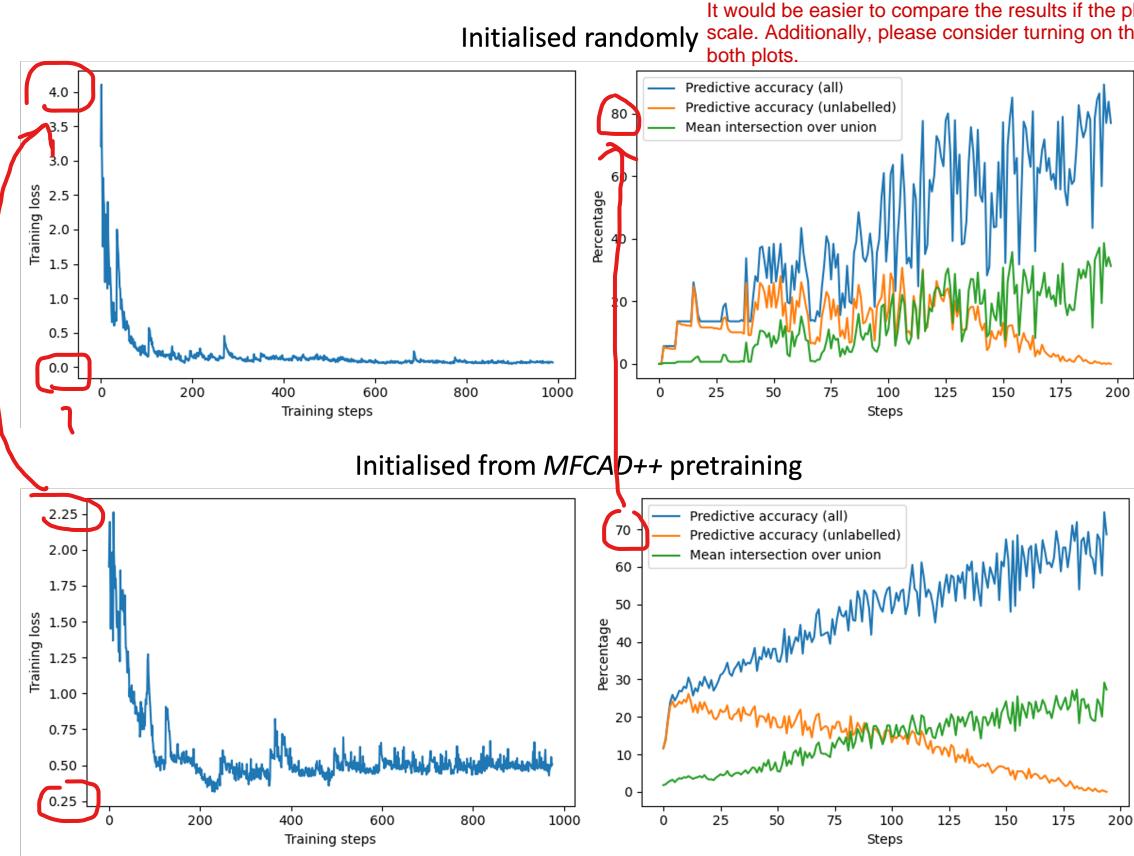


Fig. 5 Line plots showing the learning progress of individual models on an arbitrary fuel spray nozzle.

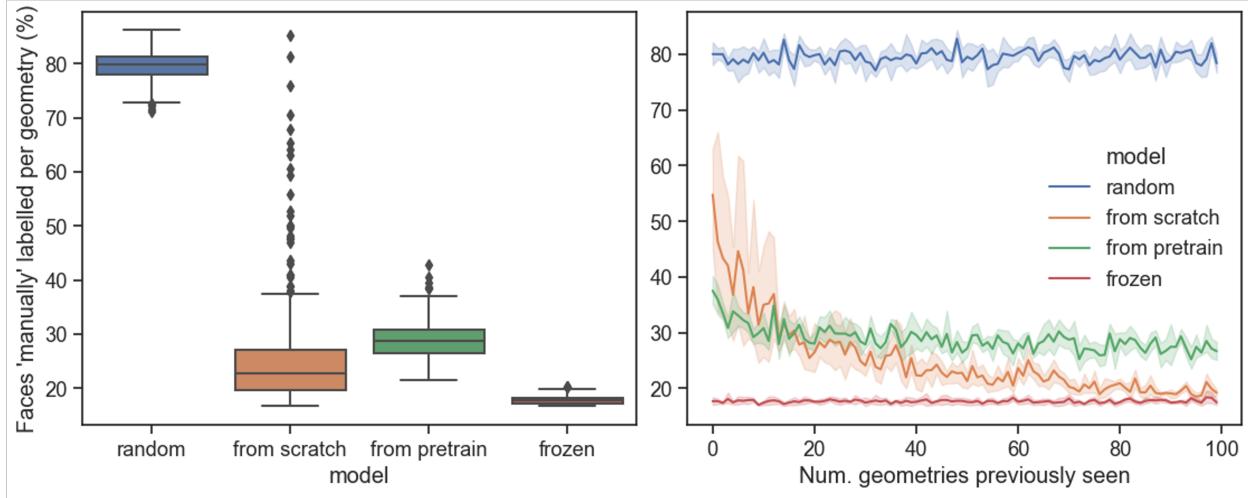
One can see that the amount of ‘manual labelling’ that the incremental learning configurations offload is somewhere in between a random baseline and a properly trained model. It is observed that the pretraining causes the models to be more robust to diverse geometries, suggested by the tighter distributions.

It is also observed that the distribution for the models initialised randomly (from scratch) when learning on the fuel spray nozzles is shifted down compared to the pretrained models. It is speculated that this is from the lack of bias. In other words, the pretrained models may be starting at a local minimum in the loss function which was good for the pretraining dataset but not as appropriate for the target dataset. Figure 5 supports this hypothesis. Here the learning progress from a single run and single geometry has been visualised. One can see that the pretrained model starts with a lower loss but the randomly initialised model eventually converges to a lower asymptotic value. This effect is observed in the fuel spray nozzles test case but not the combustion chamber liners test case likely because of the difference in geometry size and thus more opportunity to learn. [More analysis of single geometry learning will be presented in the final manuscript.]

B. Persistent learning from sequence of geometries

Similar to [18, 25], this work also analyses the behaviour of the weakly supervised incremental learning approach when continuously learning across a sequence of geometries. This experiment is similar to above, however the learning configurations do not have their weights reset between each geometry and thus retain the information learned.

Figures 6 and 7 show the results when continuously learning on each engineering test case. The same overall trends identified in subsection VI.A are present but the learning configurations perform slightly better. In addition, the distributional shift of the models initialised randomly relative to the pretrained models is greater. The hypothesis posed in the previous subsection is further reinforced with the line plots in figures 6 and 7. The pretrained models have an advantage in the beginning but seem to be constrained and are later surpassed as the randomly initialised models are further trained. It is interesting that these configurations approach the baseline ideal value of the ‘frozen’ configuration.



again, keep the scales the same and turn on minor grid lines for Fig. 6 and Fig. 7

Fig. 6 Combustion chamber liners dataset. (Left) Boxplots of distributions made by different model configurations. (Right) Line plot of performance metric achieved for each geometry in the sequence. Mean line is shown with 95% confidence intervals.

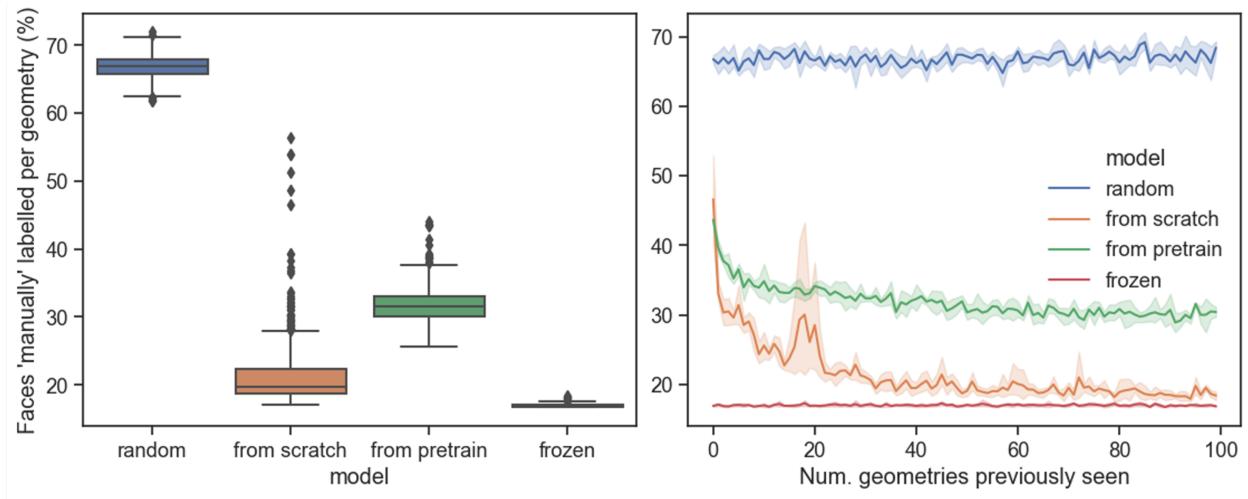


Fig. 7 Fuel spray nozzles dataset. (Left) Boxplots of distributions made by different model configurations. (Right) Line plot of performance metric achieved for each geometry in the sequence. Mean line is shown with 95% confidence intervals.

The advantage of the pretrained models is minimal in figure 7. This is likely explained by the large geometries. As suggested in figure 5, the fuel spray nozzle has enough faces (over 400) that the NNs are significantly trained. can we see this directly from Fig. 5?

[Further stochastic runs are planned to make the confidence intervals more robust for the final manuscript. Currently, only 5 runs for each configurations have been completed. More fine-grained analysis is also underway to investigate the intricacies of learning. Specifically, what characteristics of individual geometries and geometry transitions cause spikes in performance. These will be presented in the final manuscript.]

VII. Conclusion

This work suggests that weakly supervised incremental learning can produce models that provide useful suggestions to a user to offload manual labelling and accelerate the task. Both training from random initialisation and transfer learning from a public dataset significantly improves a random baseline even when learning within a single geometry. It was also shown that continuous learning across a sequence of geometries improves overall performance. Results suggest that as the length of the interaction increases (number of geometries seen), the performance of models trained from scratch approaches that of a frozen model trained with standard fully supervised learning. Interestingly, preliminary results might suggest that pretraining does not give a significant advantage in this learning scenario [although this will be investigated further for the final manuscript]. This work serves to apply the developing research on user interactive learning in image segmentation to the CAD space. A proof-of-concept is presented to illustrate that a NN is able to learn a new label space with weakly supervised incremental learning. [Conclusions will be finalised for the final manuscript].

Acknowledgments

The authors gratefully acknowledge support from the UK Defence Science and Technology Labs via contract DSTLX-1000152302, and special thanks to Dr. Fred Witham for his support in this research.

References

- [1] Ricard, S., “AI’s Effect On Productivity Now And In The Future,” *Forbes*, 2020. URL <https://www.forbes.com/sites/forbestechcouncil/2020/03/20/ais-effect-on-productivity-now-and-in-the-future/>, accessed on 2023-05-10.
- [2] Willis, K., “How Machine Learning Can Help Shape the Future of Design,” *Autodesk Blog*, 2020. URL <https://www.autodesk.com/products/fusion-360/blog/how-machine-learning-can-help-shape-the-future-of-design/>, accessed on 2023-05-10.
- [3] Bass, J., “AI for Real Value Add,” *Siemens Blog*, 2021. URL <https://blogs.sw.siemens.com/nx-design/ai-for-real-value-add/>, accessed on 2023-05-10.
- [4] Pagliarini, C., “AI Cannot Replace Creativity but It Can Help,” *The SOLIDWORKS Blog*, 2023. URL <https://blogs.solidworks.com/solidworksblog/2023/01/save-time-on-tedious-tasks-with-ai-features-in-3d-creator.html>, accessed on 2023-05-10.
- [5] Shah, J. J., Anderson, D., Kim, Y. S., and Joshi, S., “A Discourse on Geometric Feature Recognition From CAD Models,” *Journal of Computing and Information Science in Engineering*, Vol. 1, No. 1, 2000, pp. 41–51. <https://doi.org/10.1115/1.1345522>, URL <https://doi.org/10.1115/1.1345522>.
- [6] Zhang, X., Toal, D. J., Bressloff, N., Keane, A., Witham, F., Gregory, J., Stow, S., Goddard, C., Zedda, M., and Rodgers, M., “Prometheus: a geometry-centric optimisation system for combustor design,” *ASME Turbo Expo 2014: Turbine Technical Conference and Exposition (15/06/14 - 19/06/14)*, 2014. URL <https://eprints.soton.ac.uk/363186/>.
- [7] Thakur, A., Banerjee, A. G., and Gupta, S. K., “A survey of CAD model simplification techniques for physics-based simulation applications,” *Computer-Aided Design*, Vol. 41, No. 2, 2009, pp. 65–80. <https://doi.org/10.1016/j.cad.2008.11.009>, URL <https://www.sciencedirect.com/science/article/pii/S0010448508002285>.
- [8] Colligan, A., Robinson, T., Nolan, D., Hua, Y., and Cao, W., “Hierarchical CADNet: Learning from B-Reps for Machining Feature Recognition,” *Computer-Aided Design*, Vol. 147, 2022. <https://doi.org/10.1016/j.cad.2022.103226>.
- [9] Vidanes, G., Toal, D., Zhang, X., Keane, A., Gregory, J., and Nunez, M., “Extending Point-Based Deep Learning Approaches for Better Semantic Segmentation in CAD,” *Computer-Aided Design*, ????. Unpublished, accepted.

- [10] Zhang, Z., Jaiswal, P., and Rai, R., “Featurenet: Machining feature recognition based on 3d convolution neural network,” *Computer-Aided Design*, Vol. 101, 2018, pp. 12–22.
- [11] Shi, P., Qi, Q., Qin, Y., Scott, P. J., and Jiang, X., “A novel learning-based feature recognition method using multiple sectional view representation,” *Journal of Intelligent Manufacturing*, Vol. 31, No. 5, 2020, pp. 1291–1309.
- [12] Yao, X., Wang, D., Yu, T., Luan, C., and Fu, J., “A machining feature recognition approach based on hierarchical neural network for multi-feature point cloud models,” *Journal of Intelligent Manufacturing*, 2022, pp. 1–12.
- [13] Cao, W., Robinson, T., Hua, Y., Boussuge, F., Colligan, A. R., and Pan, W., “Graph representation of 3d cad models for machining feature recognition with deep learning,” *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 84003, American Society of Mechanical Engineers, 2020, p. V11AT11A003.
- [14] Zhang, H., Zhang, S., Zhang, Y., Liang, J., and Wang, Z., “Machining feature recognition based on a novel multi-task deep learning network,” *Robotics and Computer-Integrated Manufacturing*, Vol. 77, 2022, p. 102369. <https://doi.org/https://doi.org/10.1016/j.rcim.2022.102369>, URL <https://www.sciencedirect.com/science/article/pii/S0736584522000564>.
- [15] Lambourne, J. G., Willis, K. D., Jayaraman, P. K., Sanghi, A., Meltzer, P., and Shayani, H., “BRepNet: A Topological Message Passing System for Solid Models,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 12773–12782.
- [16] Jayaraman, P. K., Sanghi, A., Lambourne, J. G., Willis, K. D., Davies, T., Shayani, H., and Morris, N., “Uv-net: Learning from boundary representations,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11703–11712.
- [17] Bearman, A., Russakovsky, O., Ferrari, V., and Fei-Fei, L., “What’s the point: Semantic segmentation with point supervision,” *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, Springer, 2016, pp. 549–565.
- [18] Lenczner, G., Chan-Hon-Tong, A., Le Saux, B., Luminari, N., and Le Besnerais, G., “Dial: Deep interactive and active learning for semantic segmentation in remote sensing,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 15, 2022, pp. 3376–3389.
- [19] Prabhakar, S., and Henderson, M. R., “Automatic form-feature recognition using neural-network-based techniques on boundary representations of solid models,” *Computer-Aided Design*, 1992.
- [20] Sunil, V., and Pande, S., “Automatic recognition of machining features using artificial neural networks,” *The International Journal of Advanced Manufacturing Technology*, 2009.
- [21] Xu, N., Price, B., Cohen, S., Yang, J., and Huang, T. S., “Deep interactive object selection,” *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 373–381.
- [22] Le Saux, B., and Sanfourche, M., “Rapid semantic mapping: Learn environment classifiers on the fly,” *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2013, pp. 3725–3730.
- [23] Lenczner, G., Saux, B. L., Luminari, N., Tong, A. C. H., and Besnerais, G. L., “DISIR: Deep image segmentation with interactive refinement,” *arXiv preprint arXiv:2003.14200*, 2020.
- [24] Teng, E., Falcão, J. D., Huang, R., and Iannucci, B., “Clickbait: click-based accelerated incremental training of convolutional neural networks,” *2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, IEEE, 2018, pp. 1–12.
- [25] Kontogianni, T., Gygli, M., Uijlings, J., and Ferrari, V., “Continuous adaptation for interactive object segmentation by learning from corrections,” *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, Springer, 2020, pp. 579–596.
- [26] Benenson, R., Popov, S., and Ferrari, V., “Large-scale interactive object segmentation with human annotators,” *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11700–11709.
- [27] Lenczner, G., Chan-Hon-Tong, A., Luminari, N., Saux, B. L., and Besnerais, G. L., “Interactive learning for semantic segmentation in Earth observation,” *arXiv preprint arXiv:2009.11250*, 2020.
- [28] Lenczner, G., Chan-Hon-Tong, A., Luminari, N., and Le Saux, B., “Weakly-supervised continual learning for class-incremental segmentation,” *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*, IEEE, 2022, pp. 4843–4846.

- [29] Bian, S., Grandi, D., Hassani, K., Sadler, E., Borjin, B., Fernandes, A., Wang, A., Lu, T., Otis, R., Ho, N., et al., “Material prediction for design automation using graph representation learning,” *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 86229, American Society of Mechanical Engineers, 2022, p. V03AT03A001.
- [30] Jones, B., Hildreth, D., Chen, D., Baran, I., Kim, V. G., and Schulz, A., “Automate: A dataset and learning approach for automatic mating of cad assemblies,” *ACM Transactions on Graphics (TOG)*, Vol. 40, No. 6, 2021, pp. 1–18.
- [31] Qi, C. R., Yi, L., Su, H., and Guibas, L. J., “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” *Advances in neural information processing systems*, 2017.
- [32] Toal, D. J. J., Zhang, X., Keane, A. J., Lee, C. Y., and Zedda, M., “The Potential of a Multifidelity Approach to Gas Turbine Combustor Design Optimization,” *Journal of Engineering for Gas Turbines and Power*, Vol. 143, No. 5, 2021. <https://doi.org/10.1115/1.4048654>, URL <https://doi.org/10.1115/1.4048654>, 051002.
- [33] Kingma, D. P., and Ba, J., “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.