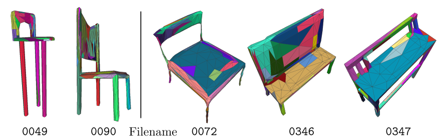
We thank the reviewers for their valuable feedback. We are pleased that they found the approach to be well-designed [R1], the expositions, illustrations, and experimentation to be comprehensive and detailed [R2, R3], the method "quite elegant" [R3], and the analysis of limitations of prior works solid [R4].

Contributions. We apologize for the lack of clarity that caused some sentences to be perceived as overclaiming; we will carefully rephrase them. R1, R2, R4: *"incrementality of approach"/"value of knowledge transfer"*: while it is true that we use the convolution from [20], we would like to stress that this was only applied on generic graph benchmarks, and that the idea of reconducting primal-dual graphs to meshes is novel. This extension, together with the idea of "bridging the gap" between graph-based ([20]) and ad-hoc methods for meshes, enables to use tools from the graph-NN literature in the context of meshes and to combine them with tools from the mesh-processing literature. Specifically, it allows implementing dynamic aggregation on meshes (via attention), and to assign features not only to edges, but also to faces. Finally, we are the first, in the context of learning-based mesh processing, to perform an attention-based pooling operation, which can be geometrically interpreted as face clustering. R2: *"geometric interpretation"*: the "uniqueness" is in its *type*, not in the fact that we have one; R4: *"network identifies high-level structure with semantic meaning"*: we apologize for the lack of clarity: our intention was not to claim that all sub-parts identified by the network have a semantic meaning, but rather that the approach can identify «semantic components in a mesh *more naturally* [...]» than other methods. (L318-320). We believe this work paves the way for a number of research avenues (L334-337): our pooling operation goes in the direction of abstracting similar larger structures across similar samples. In the Figure above (same color $\leftrightarrow$ same face cluster): on the left, the legs of the chairs are identified as clusters; on the right, larger planar structures are found. We stress that the network does not always find such structures, but also highlight that no supervision on face clusters was provided during training. To reproduce figures: see `PD-MeshNet/docs/results.md`, command `python test.py --f $UNZIP_FOLDER/coseg/chairs/ --save_clusters`.



Ablation studies. As suggested [R1, R4], we provide ablation studies to evaluate the contribution of the components. We use the Human Body dataset with same network parameters as the main paper, and train for 300 epochs. We do not perform data augmentation to further isolate the importance of each component. Table 1 shows the result of the experiments. When pooling is removed, taking away the convolution on the dual graph (*P no Pool*) worsens performance by $\sim 31\%$ w.r.t. doing primal-dual convolution (*Ours no Pool*). Adding pooling significantly increases performance (by 6.74%, cf. *Ours (mean)*). Following the suggestion of R3, we ablated the reduction function of the pooling layer and noticed a very small performance improvement when changing the aggregation from mean to sum (*Ours (sum)*). Finally, we ablate the importance of the dual-graph by removing pooling and predicting labels on dual nodes (mesh edges). Note that this accuracy is not comparable to the one in Tab. 1. Using only the dual graph results in edge-label accuracy of 80.16%; on the other hand, adding the primal graph (while keeping the training/testing procedure still on edges) produces an accuracy of 80.39%.

P no Pool	Ours no Pool	Ours (mean)	Ours (sum)
45.67%	77.53%	84.27%	**84.52%**

Table 1: Ablation Study (Face-label acc.)

Improvement over SOTA. We acknowledge that our method is slightly "more complicated" [R1] than others, but we would like to stress that we improve on the SOTA by up to 8.1% on SHREC, 2.23% on Cube Engraving and up to 5% on COSEG (cf. Table $1-3$ in main paper). R4: *"different metrics"/"re-run results"*: MeshCNN uses an ad-hoc accuracy for segmentation; please cf. Sec. G of supplementary for more details. To ensure fairness, Table G.7 in the supplementary reports results for 3 different accuracies, including the one by MeshCNN. Furthermore, while for classification we were able to reproduce the results of MeshCNN, this was partially not the case for COSEG (the experiment on the `chairs` subset gave results significantly lower than those reported in their paper). We thus decided to report the values that we were able to reproduce with the official MeshCNN code.

Technical details and clarifications. R2: *"duality"*: $\mathcal{G}(\mathcal{M})$ and $\mathcal{P}(\mathcal{M})$ are dual in the classical graph-theory sense (e.g., [33, 34]), while $\mathcal{P}(\mathcal{M})$ and $\mathcal{D}(\mathcal{M})$ are dual in the sense introduced by [20]; *"$\mathcal{G}(\mathcal{M})$ as primal graph"*: possible extension, but using $\mathcal{P}(\mathcal{M})$ as primal graph also allows defining our pooling operation: edge contraction in $\mathcal{G}(\mathcal{M})$ is instead a classical "edge collapse", as in MeshCNN, thus more limited (cf. Sec. C.1 in supplementary); R3: *"pooling"*: it is done in parallel through tensor operations; R4: *"size of test dataset"*: not a problem only for our method; we wanted to stress that the limited size of the test dataset and the «the large intra-class variability» (L326) could amplify the effect of wrong predictions; R4: *"capturing the local topology of the mesh"*: the reason is in the convolution: MeshCNN aggregates features weighted by kernels *shared by all the mesh edges*, thus not tuned on the local region on which they are applied. The convolution of our approach ([20]) implements instead an attention mechanism, by which nodes in different regions of the mesh can aggregate information differently, depending on the local properties of the mesh.

Limitations. [R3] We indeed empirically noticed a performance drop when the network is too deep (due to the larger number of parameters associated with the two graphs and the more complex optimization landscape). Moreover, pooling layers can be sensitive to their threshold parameters. Indeed, too-aggressive pooling causes a degradation of results.

Miscellaneous. We will add a discussion on the additional related work [R3] and correct typos [R2].