We thank the 4 reviewers for their useful comments.

R1:
Re recommend some polishing on the derivations : We separated our derivation two fold: one is a theoretical derivation that defines what are the differential operators. The second part elaborates on obtaining these operators using the popular package PyTorch-Geometric. In light of this remark we will improve the text in the final version.

Re the addition of comparison of wall-time between different methods: We agree. In addition to the FLOPs count we presented, we now measured the time for each block, which agrees with the FLOPs count. For instance, a forward pass of a DiffGCN block (ours) takes 58ms, while for DGCNN [24] it takes 121ms (on a Titan RTX GPU). This is in congruence with the results in Table 1, where DiffGCN and DGCNN require 61.3M and 167.78M FLOPs, respectively.

R2: We thank you for the appreciation.

R3 + R4:
Re the proposed method is not state-of-the-art on the segmentation benchmark in Table 4: We achieve SOTA on ShapeNet part-segmentation. Re S3DIS: we will rephrase the abstract to distinguish between the segmentation datasets in this aspect. We examined several papers (also following R4) and found that their results originate from different architectures, and most significantly larger networks in terms of parameters. For example KPConv used about 15M parameters while in the original submission we used 2.6M parameters. Despite the large time-consumption of these experiments, in the 6 days since the reviews were published, we managed to increase the overall 6-fold mIoU by 4%, from 56.9% to 60.9%. That is by using ResNet bottlenecks, as in KPConv, but still with about 2.6M parameters so that the training will finish on time. We will update our best result in the final version of this paper, if accepted.

Re Comparison with spectral approaches: Spectral convolutions are defined either by Fourier transform or polynomials of graph Laplacian matrix (GLM). The differences are as follows: 1) GLMs are combinatorial and not geometric - they are defined only by the connections between nodes. Shifting the nodes does not change the operator. Hence, the GLM is not a classical differential operator. 2) In our convolution we project the gradient and Laplacian, defined on an edge, to the 3 directions x, y, z. 3) Fourier transform of GLM is computationally expensive, non-local in space and learned for a specific graph-structure, i.e., it is not generalized for different graphs. 4) Using GLM polynomials, one can express the mass operator and the Laplacian (zero and first order polynomials). Higher orders result in a more spread convolution, and overall, the approach can only express symmetric matrices. The gradient operator is a non-symmetric operator, mostly used for edge detection, and cannot be expressed by spectral convolutions. As noted in [13], spectral methods lead to worse results than spatial - thus we omit it in our experimental evaluations.

R3:
Re Pooling and unpooling operations and their performance: Indeed the performance of our approach with and without pooling is comparable. However, pooling significantly improves the running time, FLOPs, and energy consumption of the network. This is highly important for various real-time applications, e.g. autonomous vehicles. Hence, pooling is an essential component to make GCNs practical.

Re interpretation/related work for Eq. (2) (defining conv kernel by Gradient and Laplacian): In section 3.3 we cite [17], showing an interpretation of convolution as gradient and Laplacian operators for structured CNNs. We also mention in this section that methods like DGCNN [24] can be seen as (non-directed) gradient methods, by nodal discretization.

Re Ablation study is shown only on one benchmark: Due to space considerations, and following [23] we showed an extensive study on one experiment. More ablation experiments will be added to the supplementary of the final version.

R4:

Re Comparison with AMG work at ICML20: Although it has a similar name, the purpose of this work is completely different than ours. It defines a network that builds AMG prolongation to solve the Poisson equation (that is, classical PDEs) on unstructured grids. It is not targeted at geometric deep learning tasks on point clouds for example.

Re Other methods for classification and part segmentation methods: Spectral methods are not effective for our needs as described in [13]. DiffPool [27] proposes pooling by learning a dense clustering matrix. This is undesired for geometric deep learning for two reasons: 1) It is impractical for graphs that include thousands of vertices. 2) Such learning is restricted to fixed input and output sizes. Our method does not suffer from these issues, and includes both novel convolution kernel and pooling methods. RS-CNN differs from our work in: 1) Our method utilizes feature derivative information, contrary to the euclidean distance in RS-CNN. 2) We operate on graphs and k-neighborhoods, while RS-CNN considers correlations between points in a spherical neighborhood in 3D. Overall, our method outperforms RS-CNN both on ModelNet40 classification and ShapeNet part-segmentation. DiffPool did not include these experiments.

Re Inaccuracies, suggestions and typos: We will correct the mentioned issues, and recheck the writing.