We thank the reviewers for their constructive feedback. We will improve the presentation according to the suggestions. Below we address some major concerns.

Q1 [R1]: Does this work generalize to non-Euclidean domains with arbitrary distance measures? Yes, both the algorithm and the theorems (with appropriate bounds depending on the metric) generalize to any metric space.

Q2 [R1]: In terms of the name, the proposed work is more "geometric" than "topological". Relation to TDA. Indeed, a persistent-homology-based loss was the first thing we tried! Unfortunately, the noisy classifier as a filter function was not very helpful. We ended up using the label as the filter function and the algorithm evolved into its current form. While we agree that geometry is very important, our guarantees rely on connectivity, and hence topology is important too. A geometry-relevant filter function for persistence is a candidate worth exploring in the future.

Q3 [R2]: What is μ in Theorem 2 (Abundancy)? We think you are referring to Theorem 5. $\mu(L(\zeta))$ is the probability measure of $L(\zeta)$, the ζ-superlevel set of η. In general $\mu(A) = \int_A f$, where f is the density.

Q4 [R2]: Should be explicit the method is generic. Generalize to the ambient feature space? Thanks for pointing this out. We will make this explicit. Yes, our method works in the ambient space with similar guarantees on purity. However, the abundancy could be limited; the largest component for label i will still be pure, but could be small as it only covers one (the largest) piece of the true region of label i (in which $\eta_i \geq 1/2$). In the ambient space, the true region of label i can be scattered and even the largest piece can be small. In a deep layer of a neural net, this is not an issue, as the network is pulling the true regions of label i together.

Q5 [R2]: Can something be said about purity or abundancy improved with iterations if the algorithm? The purity remains high throughout the iterations. The abundancy grows gradually through the iterative algorithm (as shown in the brown and blue curves in Fig. 3). As stated in Q4, the abundancy depends on the size of the largest true region of label i. This region will grow gradually as the network improves. An iterative version requires decreasing ζ carefully to 0.5 to get convergence and collect most of the pure data. We leave this for future work.

Q6 [R2, R3]: Re. Theorem 1, when the data is clean, $g(\zeta)$ reduces to 1. Thanks for pointing this out. If the original data is clean, indeed our algorithm does not (cannot) gain purity (either minimum or average). However, all our theorems hold as stated as long as the noise is not zero. This is because a) by definition the minimum purity of any impure dataset is 0 (see line 165 equation (15) in supplementary), and part 1 of the theorem applies, and b) the C_ζ in part 2 of the theorem will decrease to zero as noise goes to zero. As long as the noise is not zero, choosing a higher ζ will give us higher gains ($g(\zeta)$ and C_ζ, respectively) in the purities.

Q7 [R3]: It is unclear how the bounds depend on the variables k or δ. In our theorem, the $\forall \delta, \zeta, q$ quantifiers come *before* the $\exists N, C_1, C_0$ quantifiers, and so N itself depends on ζ and δ. Since $n \geq N$, the left hand side does depend on δ. We will write $N(\delta, \zeta, q)$, $C_1(\delta, \zeta, q)$ and $C_0(\delta, \zeta, q)$, to make this explicit. We will remove the potentially misleading word "constants" from the statement, which was put to indicate that they are independent of the distribution.

Q8 [R3]: I would have appreciated a study of the time-complexity of the algorithm. The exact construction of KNN takes $O(n^2)$ time (up to $poly(k, d)$ factors), but using c-approximate nearest neighbor methods would reduce the exponent to $(1 + 1/c^2)$. In practice, we use GPU implementation to speed up the construction. The rest (computing connected components using BFS, ζ-filtering) takes $O(kn)$. Each iteration takes about 1 second for CIFAR10 (line 233-235) and 25 seconds for clothing 1M (line 274).

Q9 [R3]: Would be better to provide more testing of the algorithm and statistical analysis of the performance. Thanks for the suggestion. We also performed unpaired *t*-test (95% significance level) on the difference between the testing accuracy on CIFAR10/100. The improvement due to our method over state-of-the-art methods is statistically significant for all noise settings. Note that for Clothing1M, since the results of baseline methods in Table 2 are copied from published works (which did not provide standard deviation), we did not perform the *t*-test.

Q10 [R4]: Uniform and pair flippings are different from symmetric and asymmetric flippings. The uniform/pair flippings are exactly the same as symmetric/asymmetric flippings in [28, 39, 43]. We call it uniform/pair to emphasize the noise generation procedure, following [13, 18].

Q11 [R4]: The number of datasets used is limited (CIFAR 10/100 and Clothing1M).

The CIFAR10/100 and Clothing1M have been widely adopted as the testbeds for studying algorithms robust to label noise. Following your suggestion, we also conduct experiments on ModelNet40 [1], which contains CAD models from 40 categories. We convert the CAD models into point clouds according to [2], and employ PointNet [2] for point cloud classification. This dataset offers a different domain from images. Due to the space, here we only list the results of some representative baselines under 0.4 uniform noise, and more results will be included in the final version.

Standard	Co-teaching	Co-teaching+	RoG	PENCIL	GCE	SL	TopoFilter
74.7 ± 1.2	82.8 ± 1.1	83.0 ± 1.2	80.0 ± 0.9	81.2 ± 1.1	83.1 ± 0.5	78.8 ± 0.5	**84.2 ± 0.6**

[1] Z. Wu, *et al.* 3D ShapeNets: A Deep Representation for Volumetric Shapes. CVPR, 2015.

[2] C. R. Qi, *et al.* PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. CVPR, 2017.

Q12 [R4]: How to choose m and k? We can use the validation set to choose m and k. When there is no validation set, we choose the burn-in milestone m as the epoch when the training accuracy's increasing trend slows down. As for k, our ablation study (Fig. (4) in the main paper and Fig. (2-4) in the supplementary) shows it is quite robust. Any value between 4 and 64 gives a good performance. This is consistent with the $k \in [\Omega(\log n), O(n)]$ bound in Thm 1.