Final Project Proposals

Geometry and Topology in Machine Learning

Inés García Redondo y Martín Carrasco Castañeda

October 2026

The goal of the projects: to understand, reproduce and evaluate

The idea of the projects is for you to face a research problem and real research papers that have been published on the topic in recent years, to draw a picture of where current research stands with respect to that given question.

Below you have some proposed projects and starting literature, but feel free to propose your own projects and to build on top of what we propose here. To propose a project instead of following a direction in this document, include a project proposal following the structure of the ones below in your email indication your project preferences.

To complete the projects, a general way of proceeding could include the following steps:

- Starting from the initial reading proposed in each project, you should *read them carefully and chase the relevant references* cited therein for your chosen research direction.
- For each paper you read, you should understand what they are trying to address and the *mathematical tools that they use in their approach.*
- Analyse the *link between their proposed theory and their experiments*, and *be critical with what they claim to achieve* and what their experiments really show.
- Lastly, you should produce a code base to *reproduce some of the results* of the papers that you are studying, check if your results match with those from the papers, and critically analyze what you are observing for your proposed reasearch question.

Please, **reach out to us if you get stuck**, if you do not understand some of the mathematical content of the papers that you are studying, if you have questions about your experimental choices, anything! We can organize a meeting where you walk us through your questions and we advise you on the best way to proceed, to the best of our abilities. On the class of November 17, we will talk to each of the groups to assess the progress you have made so far.

Deliverables and logistics

For the projects you must form **groups of 2 or 3 people**, and prepare **two deliverables**:

- A **public code repository** (10% of the total mark) containing the code that you have produced reproducing the paper. You must make sure to have good documentation so that I can clone it and run it easily on my laptop, with minimal configuration on my side, using an environment manager like `uv` (https://docs.astral.sh/uv/guides/projects/). Make sure to follow good reproducibility practices; if I cannot run it, I cannot give you full marks on this part.
- A **30-minute presentation** (90% of the total mark), where all team members must participate, covering four main parts:

    1. *Mathematical ideas and literature review of the proeblem*: what is the question that you are trying to solve? What new tools or modifications from the tools we have seen in will you be

using? Do the papers that you are reading prove any mathematical results that are needed for the experiments? Why is this tool suitable and appropriate for the task you plan to solve?

2. *Structure of your code base implementation*: how have you managed to reproduce their code? What libraries did you need to use and how? Did the authors explain all choices on their architectures/tools? What implementation choices did you need to make?
3. *Your results:* did you manage to reproduce (at least partially) the results from the experimental sections of the papers you have consulted? What experiments did you run and what results did you achieve? What are your conclusions observing these results?
4. *Limitations and future work:* do you have any criticisms for the existing work work you have been reading? Did you find any limitations for this approach? Can you think of where to take this next?

Deadlines

November 2	Last day to send the group members and three project topics in order of preference. Groups formed and projects assigned.
November 17	Mid-project assessment.
December 11	Submit your code repository by email.
December 14	Submit your presentation slides by email.
December 15	Presentations. The presentation order will be announced closer to the date.

Repository submission. Without access to your repository, we cannot assess the experiments reported in your presentation. Failure to submit it will therefore negatively affect the assessment of both deliverables.

Learning on graphs

1. Sheaf Neural Networks

Jakob Hansen and Thomas Gebhart. "Sheaf Neural Networks". In: *NeurIPS Workshop on Topological Data Analysis and Beyond*. Paper | OpenReview. 2020

Federico Barbero, Cristian Bodnar, Haitz Sáez de Ocáriz Borde, Michael Bronstein, Petar Veličković, and Pietro Liò. "Sheaf Neural Networks with Connection Laplacians". In: *Topology, Algebra, and Geometry in Machine Learning (TAG-ML) Workshops*. Vol. 196. Proceedings of Machine Learning Research. Paper. 2022, pp. 28–36

Sheaf neural networks generalize graph convolution by attaching vector spaces to nodes and edges and using linear maps to describe their relationships. Hansen and Gebhart introduce diffusion with the sheaf Laplacian, allowing richer interactions, including signed and asymmetric relations. Barbero et al. construct connection maps by aligning estimated tangent spaces of neighbouring data points.

Research direction 1: Explore one construction of the SNN (Sheaf Neural Network), i.e., with the sheaf Laplacian, and explore if extra relational information is better than with ordinary GCNs for a given task in ordinary benchmark datasets

Research direction 2: Compare a Sheaf Neural Network with another geometry-based graph method, i.e., invariant/equivariant architectures

Research direction 3: Test this architecture to address over-squashing. Does it help mitigating its effects? Why?

2. Counting substructures in graphs

Charilaos I. Kanatsoulis and Alejandro Ribeiro. "Counting Graph Substructures with Graph Neural Networks". In: *International Conference on Learning Representations (ICLR)*. Paper. 2024

Paolo Pellizzoni, Till Schulz, and Karsten Borgwardt. "Graph Neural Networks Can (Often) Count Substructures". In: *International Conference on Learning Representations (ICLR)*. Paper. 2025

These papers examine what GNNs can learn about cycles, cliques, and other small graph patterns. Kanatsoulis and Ribeiro connect message passing with random node inputs to higher-order statistical moments and construct representations for counting substructures. Pellizzoni et al. study counting beyond worst-case impossibility results, giving conditions under which GNNs can learn counts efficiently.

Research direction 1: Reproduce the cycle-counting experiment for varying graph sizes

Research direction 2: Reproduce either experiment with highly symmetric graphs, i.e., strongly regular or vertex transitive

Research direction 3: What are these substructure counts adding to the message passing paradigm? Can you incorporate them into it somehow?

3. Graph rewiring to reduce oversquashing

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. "Understanding Over-Squashing and Bottlenecks on Graphs via Curvature". In: *International Conference on Learning Representations (ICLR)*. Paper. 2022

Mitchell Black, Zhengchao Wan, Amir Nayyeri, and Yusu Wang. *Understanding Oversquashing in GNNs through the Lens of Effective Resistance*. Paper. 2023. arXiv: 2302.06835

Oversquashing occurs when information from many distant nodes must pass through narrow graph bottlenecks. The first paper relates these bottlenecks to graph curvature and proposes curvature-guided rewiring. The second uses effective resistance to measure connectivity and selects additional edges to reduce total resistance.

Research direction 1: Reproduce either of the rewiring techniques and test its effectiveness on common problems designed to showcase over-squasing.

Research direction 2: Reproduce both techniques and compare their effectivenes. Are there any cases where one is stronger than the other? Can this be explained from their mathematical properties?

Research direction 3: Research on something called "Effective Resistance Curvature". Can you link both rewiring techniques through this quantity?

Latent Representations

4. VAEs and topology

Jilles S. van Hulst, Jakub M. Tomczak, W. P. M. H. Heemels, and Duarte J. Antunes. *Constructing VAE Latent Spaces with Prescribed Topology*. Preprint. Paper. 2026. arXiv: 2606.07058v1

A Gaussian latent space can mismatch data with periodic or other non-Euclidean structure. This paper constructs VAE latent spaces from elementary factors, such as circles and intervals, and their quotients, while retaining tractable training objectives. It tests topology-matched priors on synthetic manifolds and transformed MNIST images.

Research direction 1: Reproduce the periodic-data experiment and use persistent homology as an evaluation tool to assess the results.

5. Topological Autoencoders

Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. "Topology-Preserving Deep Image Segmentation". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Paper. 2019

Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. "Topological Autoencoders". In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. Paper. 2020, pp. 7045–7054

Mattéo Clémot, Julie Digne, and Julien Tierny. "Topological Autoencoders++: Fast and Accurate Cycle-Aware Dimensionality Reduction". In: *IEEE Transactions on Visualization and Computer Graphics* 32.2 (2026). Paper | arXiv | Code, pp. 1622–1639

Pixelwise segmentation losses may overlook broken connections or spurious holes. TopoAE introduces a differentiable loss based on persistent homology to encourage predictions with the topology of the ground-truth, focusing on 0-dimensional homology. TopoAE++ extends the topological-autoencoder formulation to better preserve loops in two-dimensional embeddings.

Research direction 1: Reproduce the experiments from TopoAE++ and study the obtained results. Is there any other dataset where a topology aware dimensionality reduction could be enlightening?

Research direction 2: Adapt the loss or regularizers proposed in other kinds of architectures, for example, based on GNNs or GANs. Can you think of a potential application in those cases?

Research direction 3: Evaluate and compare in which case each loss or regularizer is better

6. Graph representations using Laplacians

Junru Zhou, Cai Zhou, Xiyuan Wang, Pan Li, and Muhan Zhang. *Towards Stable, Globally Expressive Graph Representations with Laplacian Eigenvectors*. Preprint, 2024; revised 2026. Paper. 2026. arXiv: 2410.09737

Maximilian Krahn, Lennart Bastian, Vikas Garg, Björn Schuller, and Tolga Birdal. "Collapsed Effective Operators for Higher-Order Structures". In: *International Conference on Machine Learning (ICML)*. Paper. 2026. arXiv: 2606.23517v1

Laplacian-based representations provide information beyond a node's local neighbourhood. Zhou et al. construct spectral representations that account for eigenvector symmetries and handle nearby eigenvalues smoothly. Krahn et al. use Schur complementation to condense higher-order structure into a vertex-level operator, supporting clustering, smoothing, and neural positional encodings.

Research direction 1: Compare the representations with other methods of obtaining embeddings from graphs

Research direction 2: Analyze what happens in the case of co-spectral non-isomorphic graphs

Persistent Homology and Learning

7. Learning from persistence diagrams with PersLay

Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. "PersLay: A Neural Network Layer for Persistence Diagrams and New Graph Topological Signatures". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. vol. 108. Proceedings of Machine Learning Research. Paper. 2020, pp. 2786–2796

Sebastian Damrich, Philipp Berens, and Dmitry Kobak. "Persistent homology for high-dimensional data based on spectral methods". In: *Advances in Neural Information Processing Systems* 37 (2024). Paper

PersLay learns vector representations of persistence diagrams so that they can be used within neural networks. The paper also builds stable graph signatures using heat-kernel filtrations and extended persistence, and evaluates them on graph classification. The second paper evaluates different metric for building the filtrations on high-dimensional data.

Research direction 1: Compare PersLay to modern datasets and models, see if it holds the test of time

Research direction 2: Can you think of a different task or data type (e.g. images) where we could apply this architecture and find it useful?

Research direction 3: Compare PersLay with different metrics for constructing the filtrations on high-dimensions. How does the choice of the filtration affect the performance?

8. Adding persistent homology to graph neural networks

Max Horn, Edward De Brouwer, Michael Moor, Yves Moreau, Bastian Rieck, and Karsten Borgwardt. "Topological Graph Neural Networks". In: *International Conference on Learning Representations (ICLR)*. Paper. 2022

This paper introduces TOGL, a layer that incorporates persistent-homology information into a GNN. It uses graph topology to capture global structure, such as connected components and cycles, that ordinary message passing can miss, and establishes increased expressive power.

Research direction 1: Reproduce the experiments and perform a comparison between using TOGL and not using TOGL for these and new tasks

Research direction 2: How does TOGL behave with respect to over-smoothing and over-squasing?

Intrinsic dimension of internal representations

9. Local intrinsic dimension of representations in LLMs

Benjamin Matthias Ruppik, Julius von Rohrscheidt, Carel van Niekerk, Michael Heck, Renato Vukovic, Shutong Feng, Hsien-chin Lin, Nurul Lubis, Bastian Rieck, Marcus Zibrowius, et al. "Less is more: Local intrinsic dimensions of contextual language models". In: *Advances in Neural Information Processing Systems* 38 (2026). Paper

In this paper, the authors present a local variant of the TwoNN dimensionality estimator that we saw in class, and apply it to several questions related to LLMs. They investigate fine-tuning, grokking, the exhaustion of the learning capabilities and overfitting through the lens of intrinsic dimension.

Research direction 1: Reproduce the fine-tuning experiments, using LoRA and other datasets to finetune. Do you also observe the conclusion from Figure 1 in other models and datasets?

Research direction 2: Grokking is known to happen for more models and tasks than the modular arithmetic heuristic describe in the paper, see this study for more detail. Can you extend their grokking experiments to other architectures and datasets? What happens with the intrinsic dimension in those cases?