# **Statement of Work**

## **Tony Liu (RA) Hypernets Project OCT 12, 2020**

Hypernets is a research project advancing the theory and application of hypergraphs as models for complex multiway data. The primary DEC 2020 deliverable for Hypernets is an updated release of PNNL’s open sourced Python library, HyperNetX (HNX), a collection of methods and classes for the analysis of data modeled as a hypergraph. The new version will include optimized methods written in C++ and bound to a Python namespace for seamless integration into the existing framework. The optimized framework will support the practical use of HNX on very large datasets, increasing its usefulness to our sponsor.

RA will report to Tech Lead Brenda Praggastis as supervisor, backed up by Task Lead Cliff Joslyn and Project Manager Brian Kritzstein. RA’s academic advisors Prof. Gebremedhin (WSU) and Prof. Lumsdaine (UW) will be fully informed and consulted. RA will coordinate with project practices for software management and development (e.g. stash) and (virtually) attend project meetings.

The following task list and deliverables are an initial consideration, to be negotiated flexibly with the RA, Praggastis, Joslyn, and academic advisors, to be revisited as conditions warrant.

**TASKS**

1. Consult with project staff on:
   1. Efficient data structures for static (immutable) hypergraphs with core functions supported by the C++ backend.
   2. PyBind module linking the C++ backend to a Python namespace.
2. Code hypergraph algorithms in C++ for static hypergraphs. Specifics TBD, examples of target algorithms include collapsing nodes and/or edges, sub-edge reduction and closure, and s-overlap, s-distance, and related hypernetwork methods like s-connected components and s-shortest paths
3. Replace current NetworkX dependencies in HNX hypernetwork science algorithms for static hypergraphs with above C++ methods. Specifics TBD, example of target algorithms include s-centralities and s-eccentricity
4. Time permitting, replace the current Python implementation for computing mod2 homology groups and Betti numbers with optimized C++ methods.

**DELIVERABLES**

1. C++ code for hypergraph algorithms and data structures.
2. Revised HNX Python code for C++ back end.
3. C++ code for mod2 homology groups.

**LEVEL OF EFFORT**

20 hrs/wk