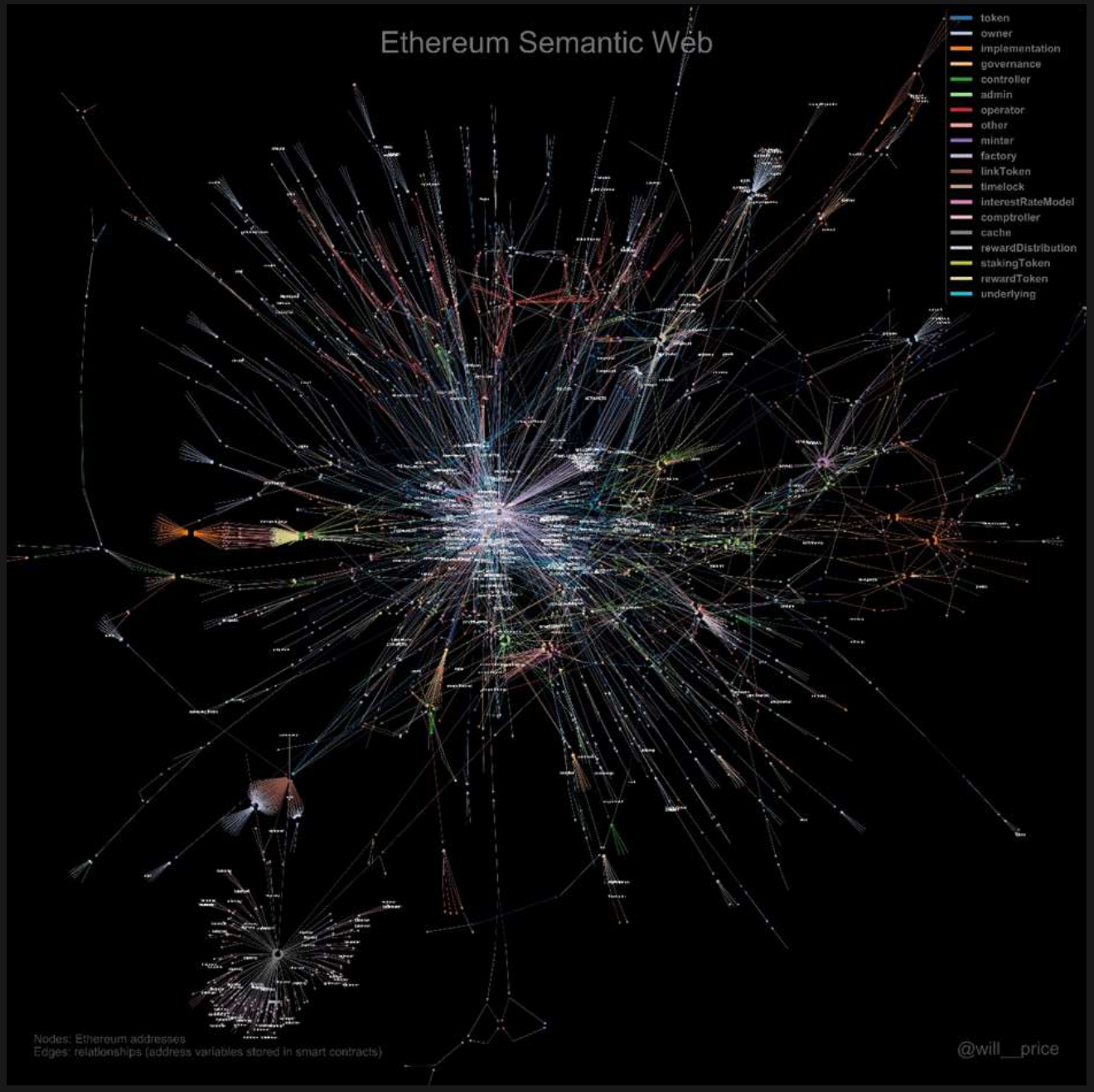
Ethereum Semantic Web
token
owner
implementation
governance
controller
admin
operator
other
minter
factory
linkToken
timelock
interestRateModel
comptroller
cache
rewardDistribution
stakingToken
rewardToken
underlying
Nodes: Ethereum addresses
Edges: relationships (address variables stored in smart contracts)
@will__price

Graph Semi-Supervised Learning (GSSL) Semi-supervised label propagation using social network analysis for smart contracts on the Ethereum network

STRUCTURE OF THE PRESENTATION

- Problem statement
- Motivation
- Current methods
- My proposal
- Current progress
- Questions

PROBLEM STATEMENT

There has been very little research into how compiled code connects to other pieces.

- In terms of their similarity in code
- Programmable interactions that link smart contracts across the EVM
- Nothing that focuses on understanding code being deployed to a system like the EVM
- no attempt at classifying smart contracts on a graph has been attempted yet

MOTIVATION

1. Classification of the long tail of Ethereum smart contracts
2. Extraction of the interconnectedness held within the network

CURRENT METHODS

There is very little research focusing on how compiled code modules can be converted into a graph of relationships. Especially in the social way that smart contracts work, with the key difference being able to call any function of any other smart contract.

MY PROPOSAL

MAIN SOURCES OF LINKS

- Use 0 class classification for every label
- Look at the code of the contract to see what it directly references
  IF THERE IS TIME / IS NEEDED
- Look at the stack trace for transactions
- Look at the storage of the contract

CURRENT IMPLEMENTATION AND NEXT STEPS

- Scraping of contracts
- Scraping of tags
- Extraction of addresses in code

ANY QUESTIONS?