Circuit Design Completion using Graph Neural Networks

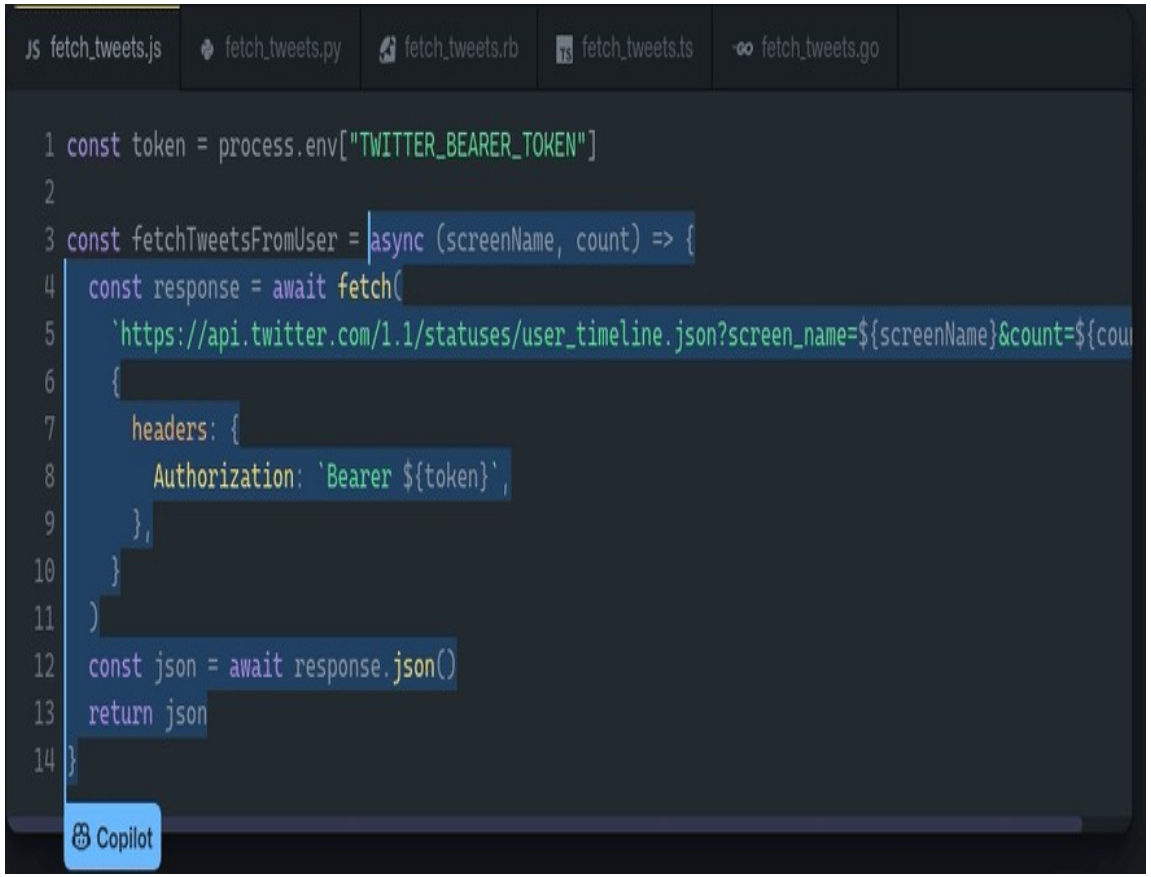
Anwar Said, Mudassir Shabbir, Brian Broll, Waseem Abbas, Peter Volgyesi, Xenofon Koutsoukos

Agenda

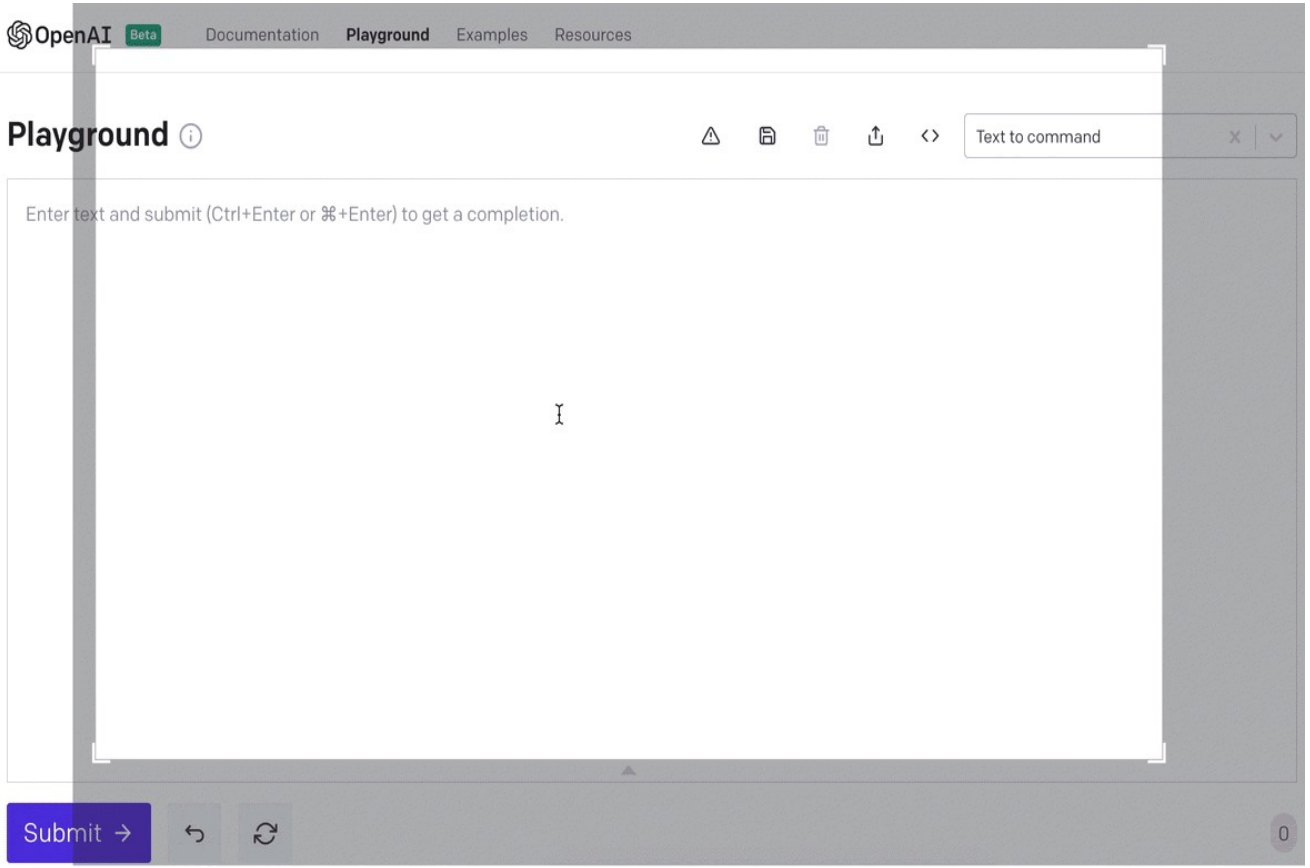
- Motivation
- Problem formulation
- Dataset generation
- Proposed methodology
- Results

Modern Innovative Tools

GitHub Copilot: Coding suggestions

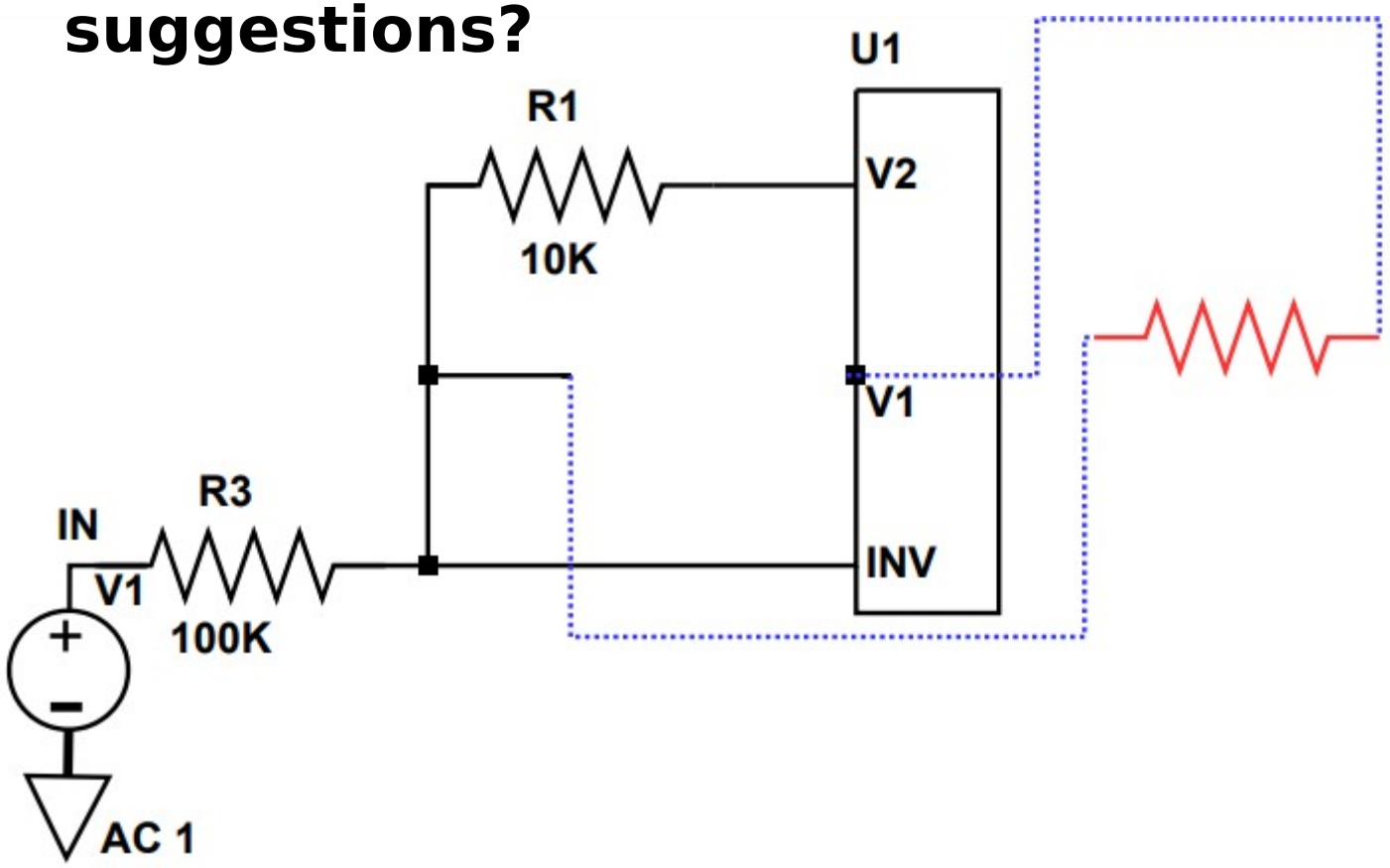


GPT3: Text completion



Circuit Design Completion

Circuit design suggestions?



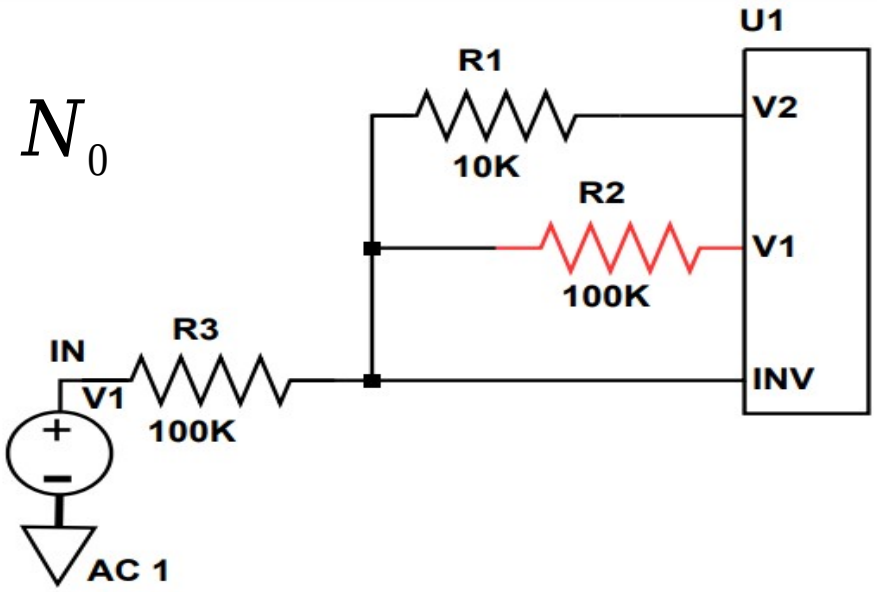
Challenges:

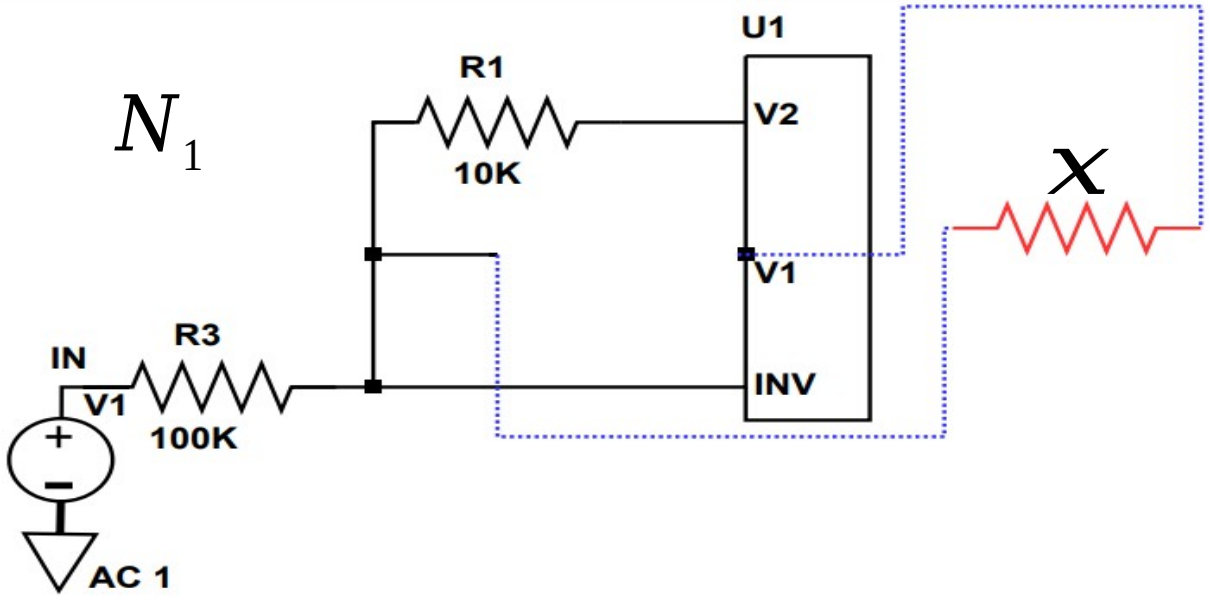
- Time consuming process due to very large search space
- Chance of errors in chips or boards after manufacturing

- **Solution**
- **ML in Circuit Design**

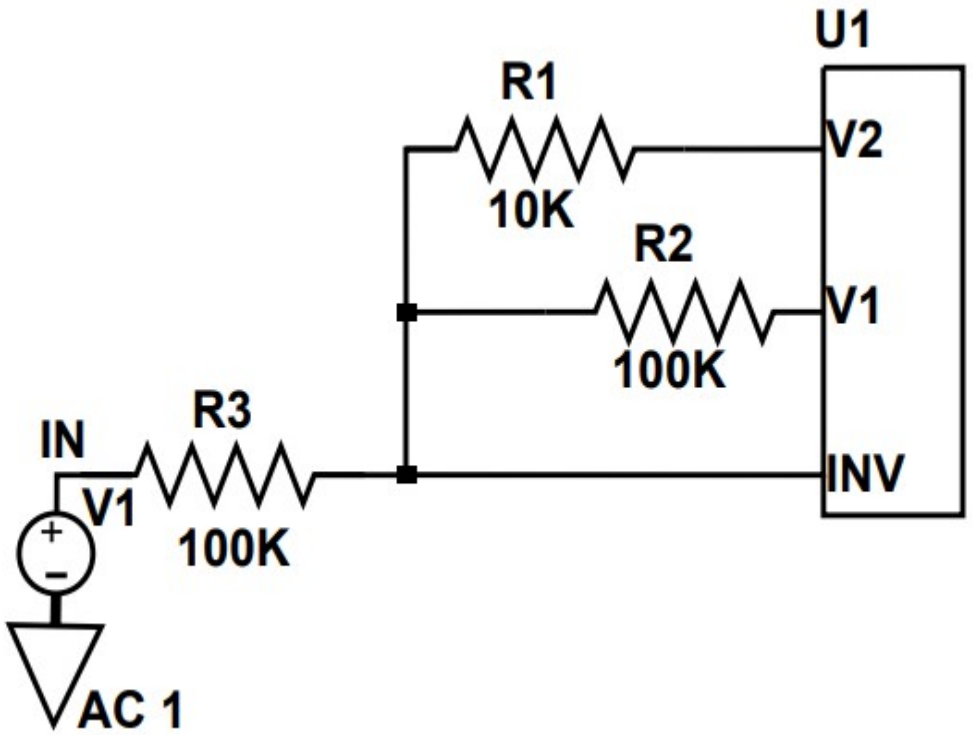
Circuit Completion Problem (CCP)

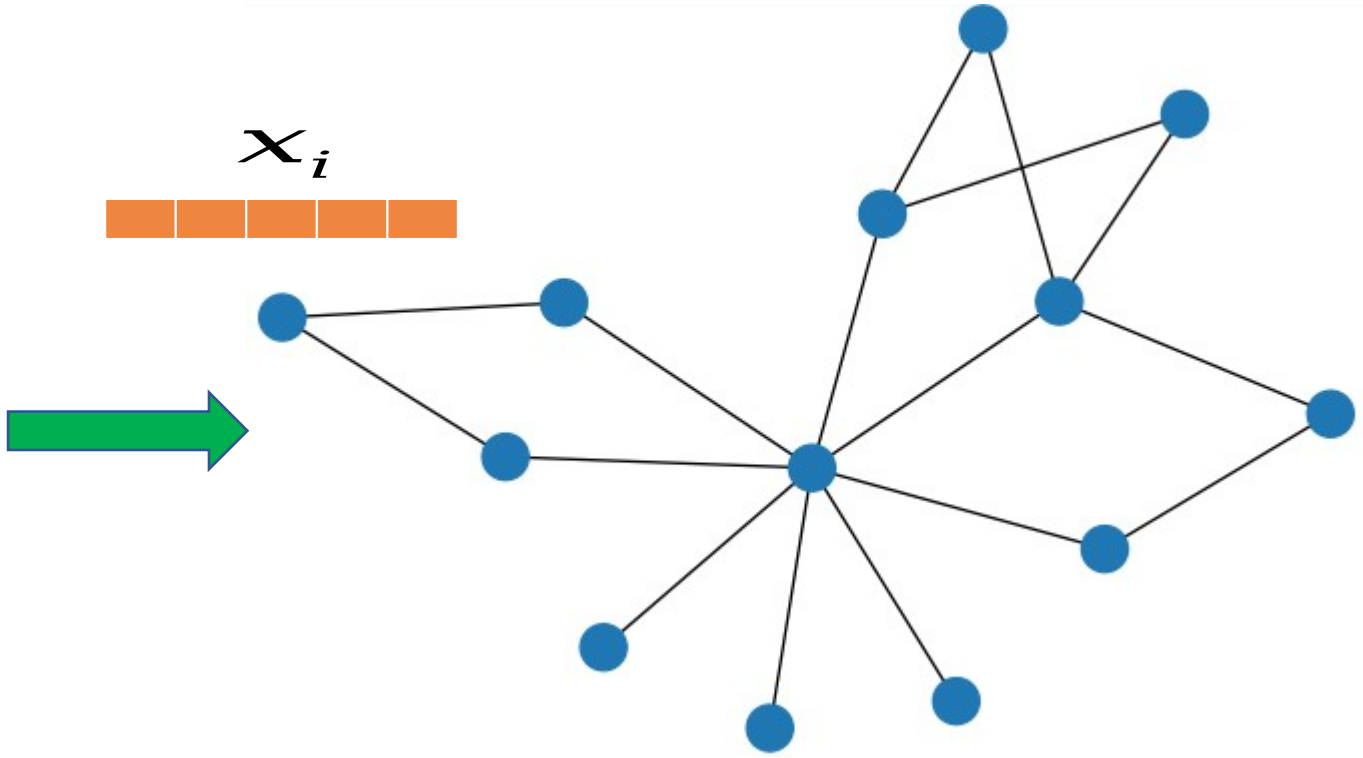
Let and , be the netlists of two circuit schematics where is an original circuit while is a copy of that is missing a component and all its connections that are present in . The circuit completion problem is to predict the type of component and all its missing connections.



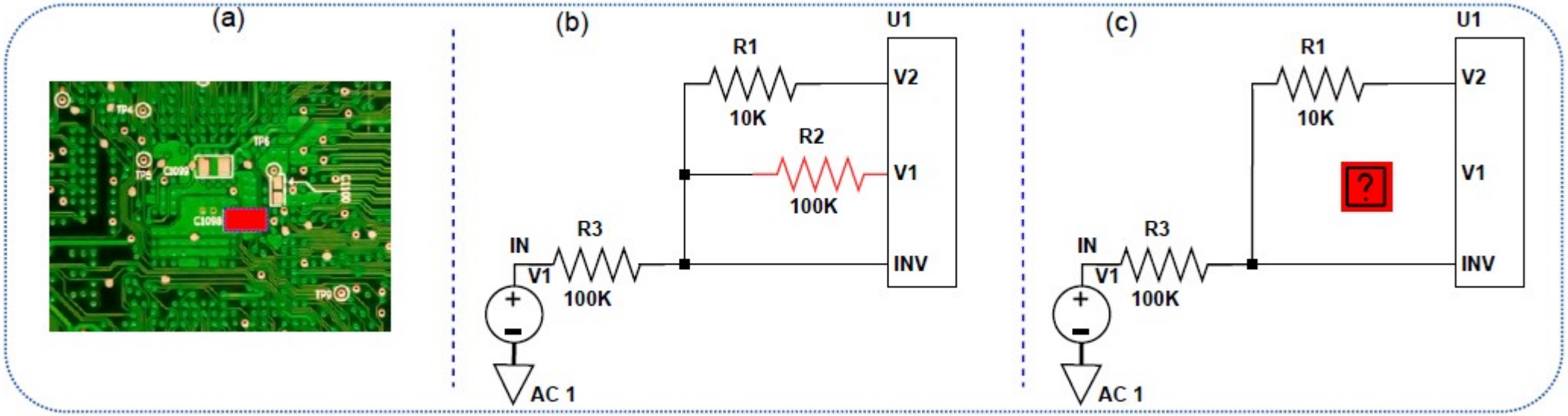


Circuit Completion using networked representation

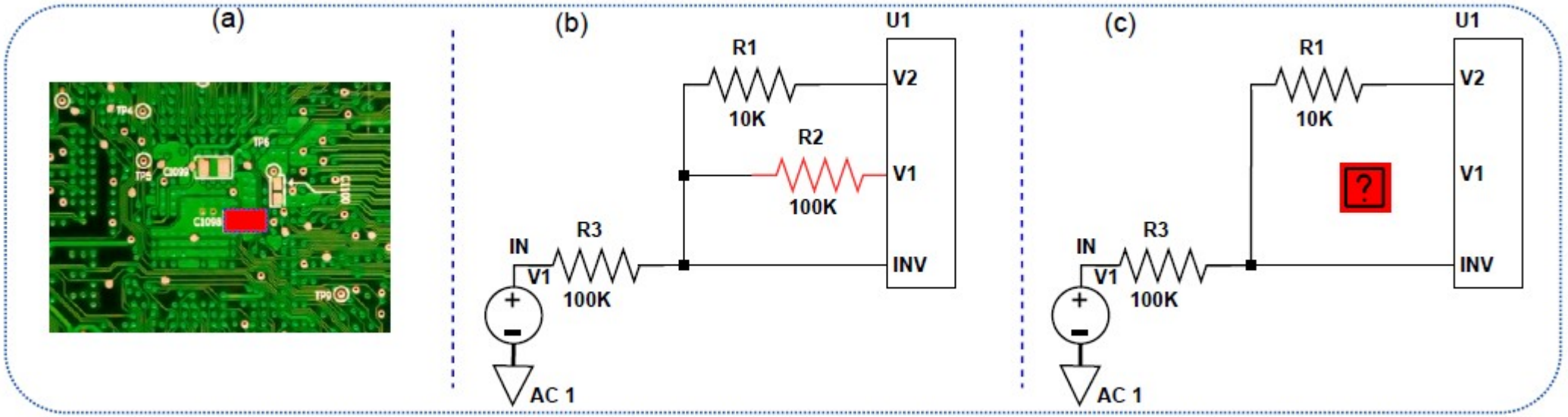




Missing Component Identification (Problem 1)

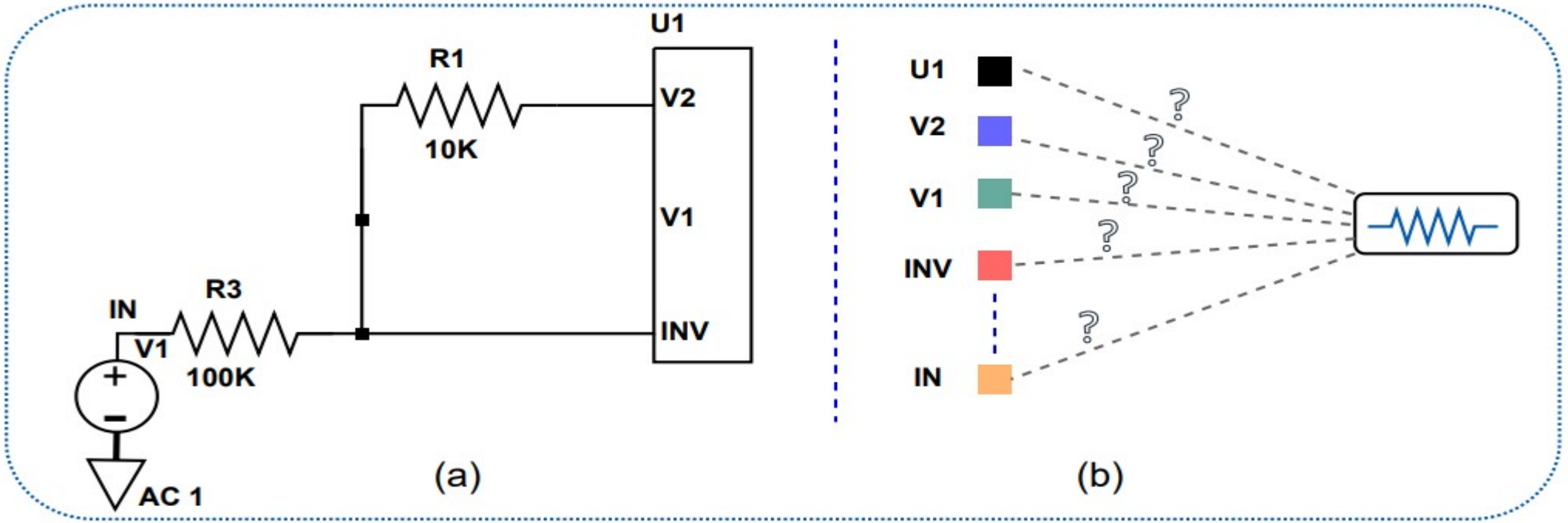


Missing Component Identification (Problem 1)

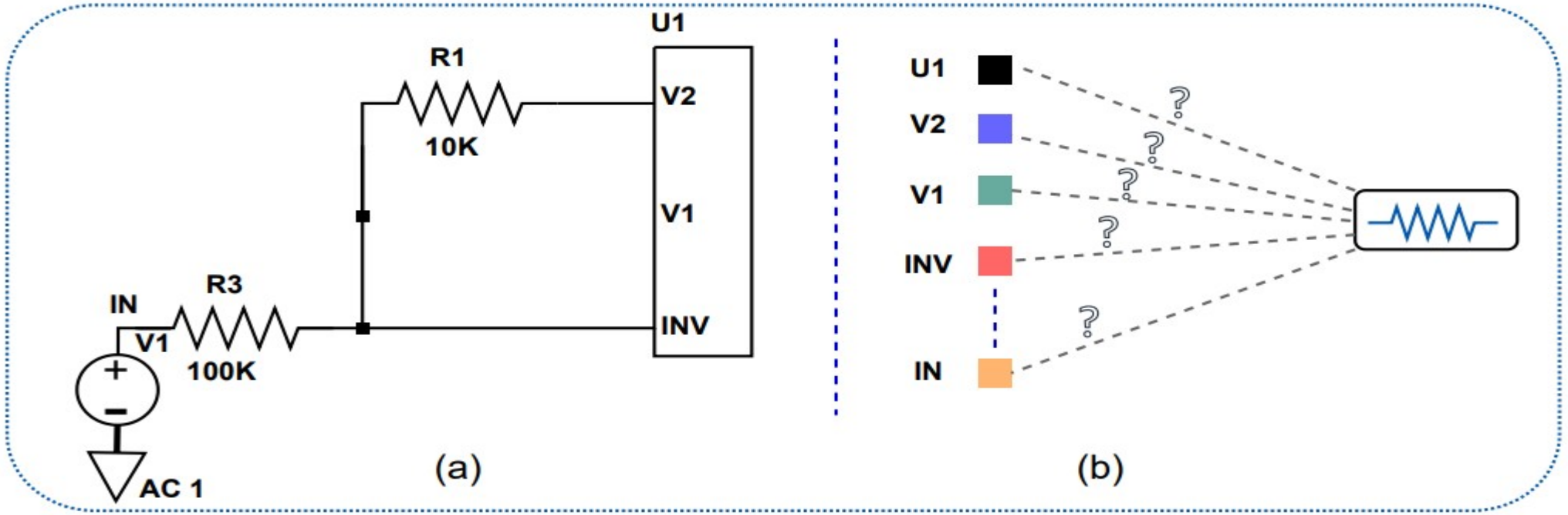


Let be a graph of a netlist with . Let be a graph obtained by removing an arbitrary vertex from . Compute the value of from .

Link Completion (Problem 2)



Link Completion (Problem 2)



Let , be a graph of a netlist with where Let be a graph obtained by removing an arbitrary vertex from . Given the value of and the , predict the set of neighbors of in

Proposed Framework

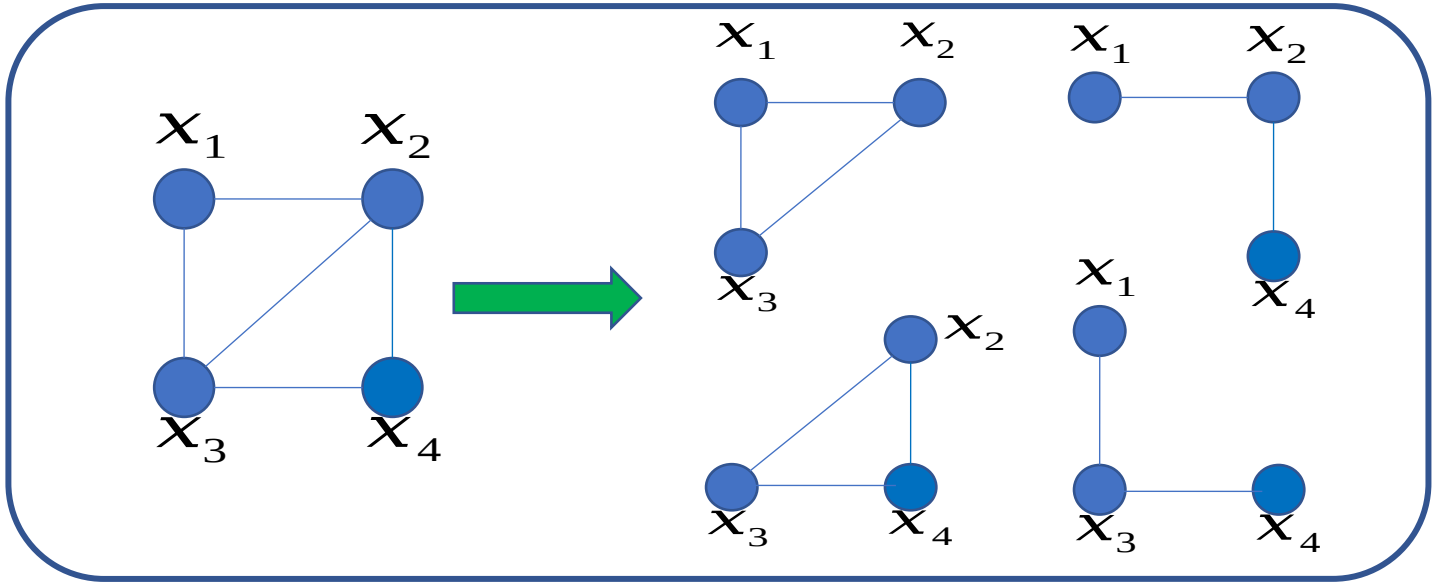
- Two ingredients:
  - Leverage Graph Neural Networks (GNNs) framework in a graph classification setting to predict missing component (problem 1)
  - Adapted GNNs based link prediction approach to predict edges for the missing components (problem 2)

Datasets

Dataset	Valid netlists	Total comp	Max comp per class	Min comp per class
Ltspice Examples	1505-377	19-18	1505-377	[1,1,6,6,8,9]
Ltspice Demos	235-59	17-12	235-59	[1,2,3,3,4,7]
Kicad Github	553-139	22-20	553-153	[2,3,3,4,12,13]

Dataset Generation (Graph Classification)

- For each graph (netlist), we remove a node for G and label the resulting graph as
- We can have as many training or testing instances as there are types of components in a netlist
- We consider only five most common component types in a given dataset

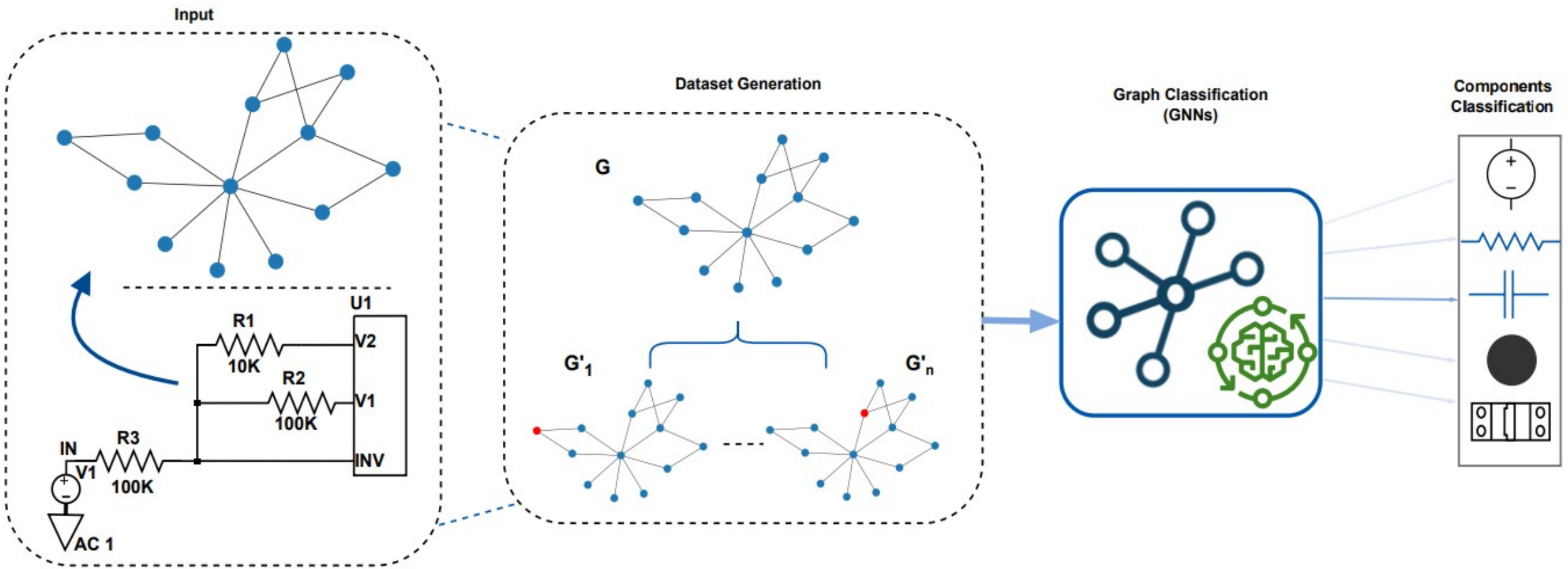


Dataset Stats

Table 2 Datasets' class-wise stats for component classification task.

	Ltspice Examples		**Ltspice Demos**		**Kicad Github**	
Component Class	Train	Test	Train	Test	Train	Test
Voltage_source	1238	303	142	38	N/A	N/A
Sub_element	1224	300	142	38	N/A	N/A
Junction-node	1243	305	142	38	254	72
Resistor	1107	272	142	38	135	38
BehavioralCap	835	189	124	37	114	32
UnifDistRCLine	N/A	N/A	N/A	N/A	142	43
JunFETrans	N/A	N/A	N/A	N/A	86	25

Component Classification using GNNs



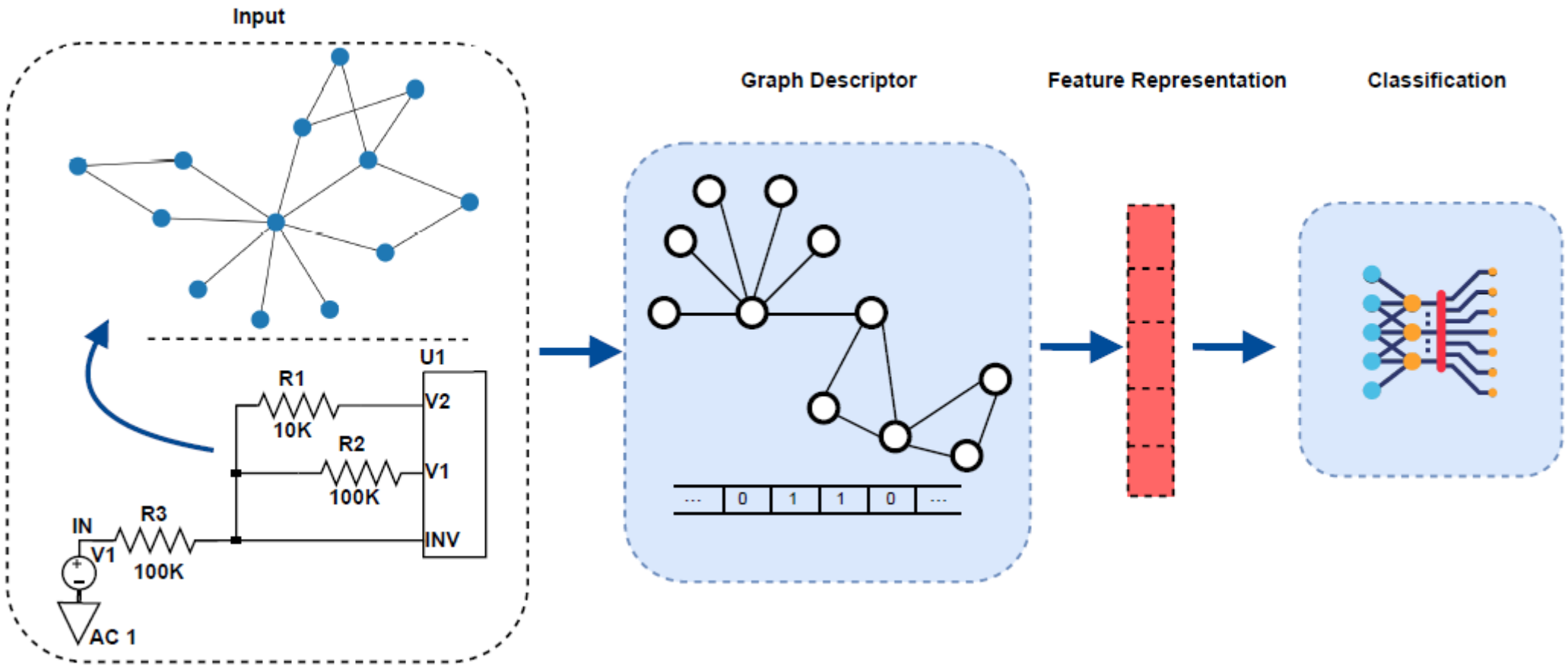
Graph Neural Networks

- Graph Convolutional Network (GCN)
- GraphSAGE
- Graph Attention Network (GAT)
- Graph Isomorphism Network (GIN)
- Nested Graph Neural Networks (NGNNs)
  - Nested GCN (NGCN)
  - Nested GAT (NGAT)
  - Nested GraphSAGE (NGraphSAGE)
  - Nested GIN (NGIN)

Evaluation measure:

Graph Descriptors

- FGSD, NetLSD, DGSD, NetSimile, WL and Shortest Path



Results

Dataset	GCN	GIN	GAT	GraphSAGE	NGCN	NGIN	NGAT	NGraphSAGE
Ltspice Examples	79.5 ± 0.01	**89.8** ± 0.01	78.1 ± 0.04	78.6 ± 0.03	81.9 ± 1.04	88.9 ± 0.01	82.7 ± 0.03	82.2 ± 0.03
Ltspice Demos	61.8 ± 0.01	**85.0** ± 0.02	58.5 ± 0.03	59.0 ± 0.04	52.6 ± 0.08	82.4 ± 0.03	53.8 ± 0.09	53.7 ± 0.06
Kicad Github	51.7 ± 0.03	**67.4** ± 0.02	56.7 ± 0.04	46.6 ± 0.02	58.1 ± 0.01	62.5 ± 0.02	56.8 ± 0.03	57.0 ± 0.02

Dataset	DGSD	NetLSD	WL	FGSD	NetSimile	Shortest Path
Ltspice Examples	77.63	77.22	77.21	48.88	**85.11**	77.21
Ltspice Demos	57.60	52.07	48.27	33.03	**70.86**	48.26
Kicad Github	47.89	47.77	49.35	51.52	**59.27**	49.35

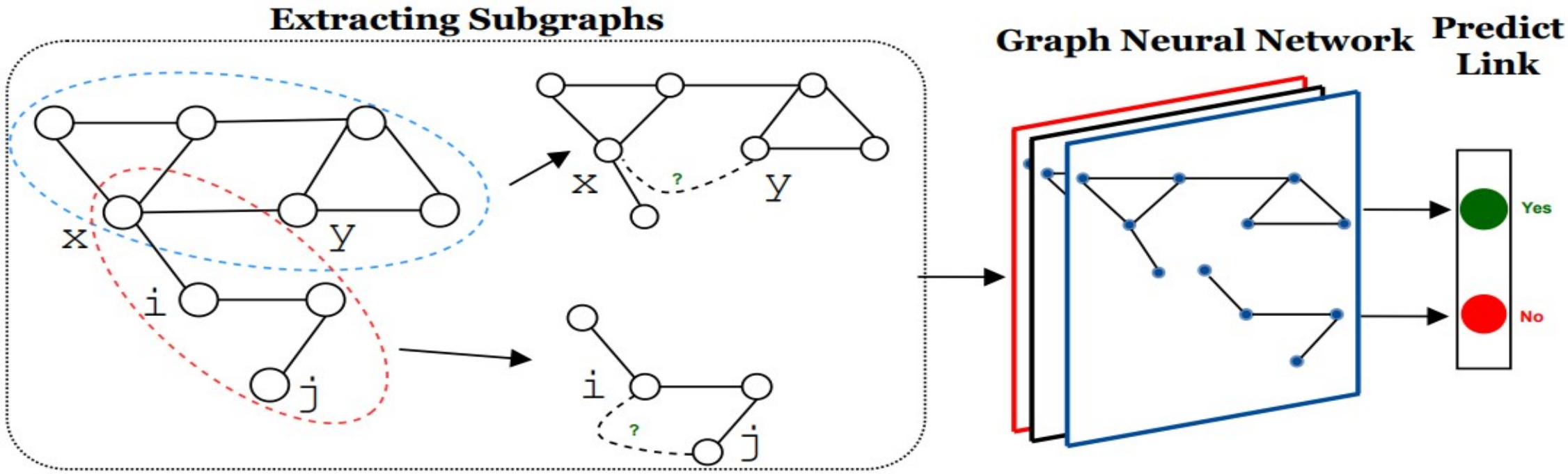
Link Completion (problem 2)

- Data preparation
  - Stack graphs and node features

$$\mathcal{A} = \begin{bmatrix} \boldsymbol{A}_1 & & \\ & \ddots & \\ & & \boldsymbol{A}_n \end{bmatrix}, \boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_n \end{bmatrix}$$

Link Completion using SEAL

- SEAL:
  - Link prediction model based GNNs
  - Extracts enclosing subgraphs around the links to extract training and testing data.



Results

Dataset	**SEAL**
Ltspice Examples	74.74
Ltspice Demos	61.39
Kicad Github	75.09

Thank you