

Typilus: Neural Type Hints

Miltos Allamanis, Earl T. Barr, Soline Ducouso, Zheng Gao



Task

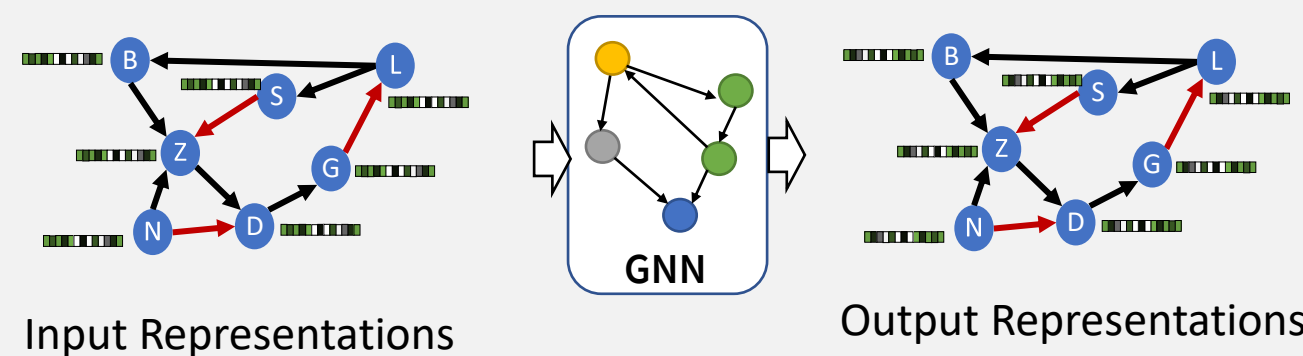
- Predict type annotations in Python with machine learning
- Allow online adaptation to new, previously unseen, types.

Why Machine Learning?

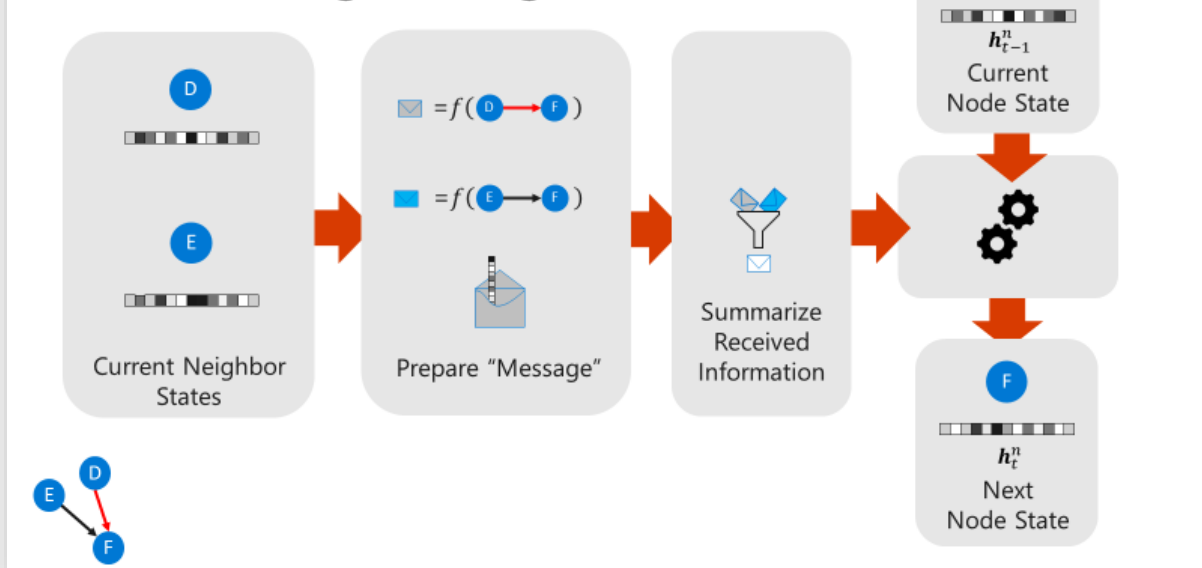
Use additional, uncertain information about programs (e.g. patterns in dataflow, names).

Graph Neural Networks (GNN)

Neural networks that operate on graph structures



Neural Message Passing



Code, Data & Tool

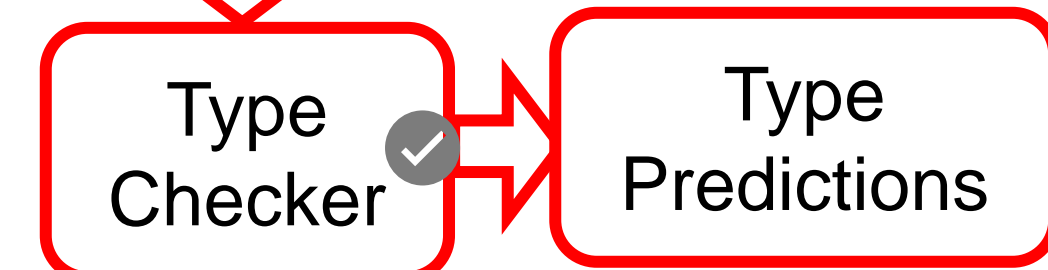
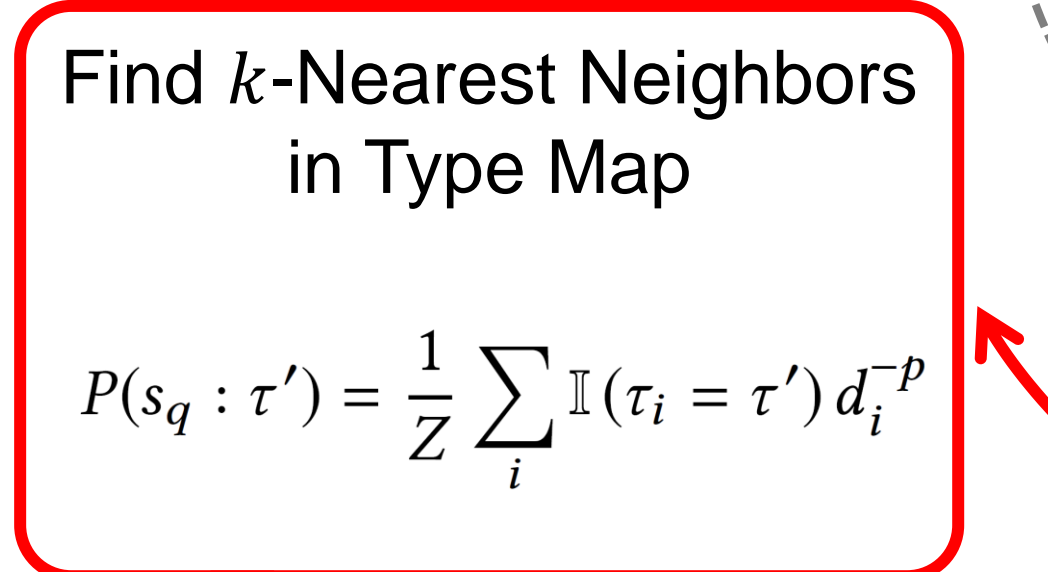
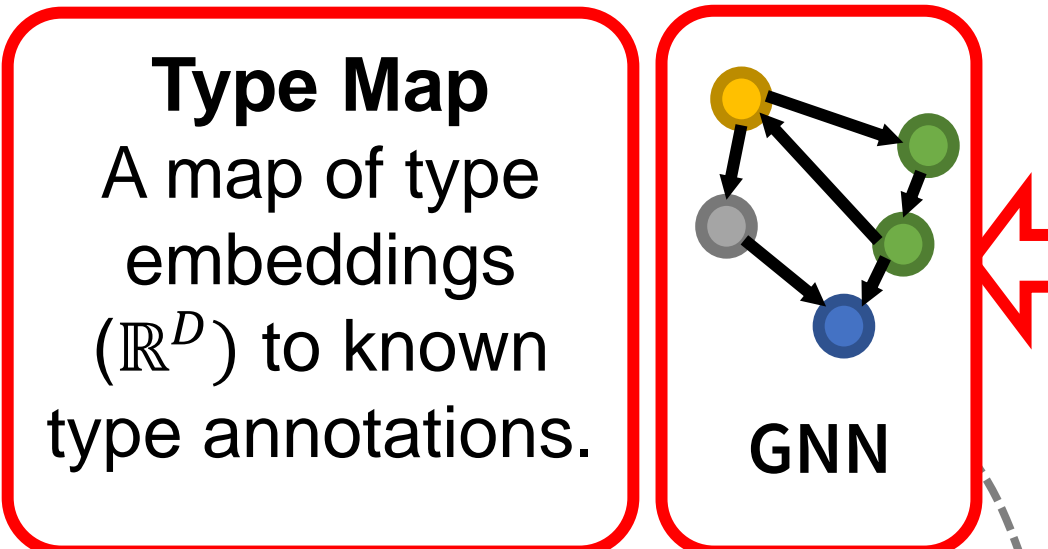
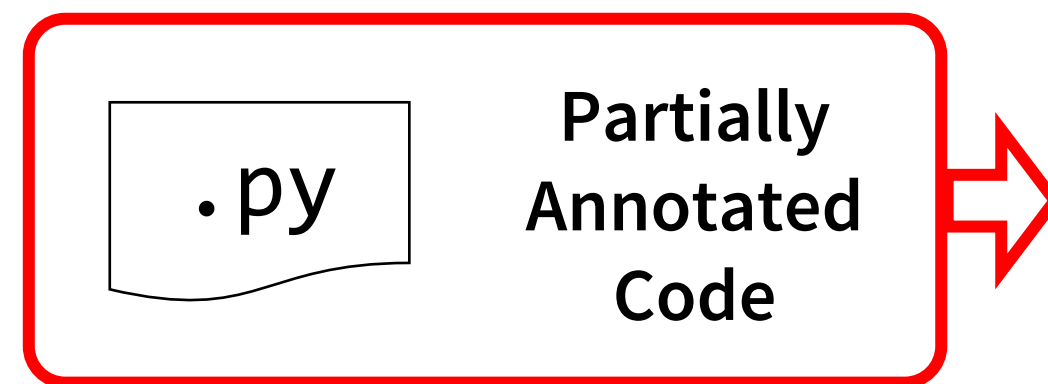


github.com/typilus/typilus

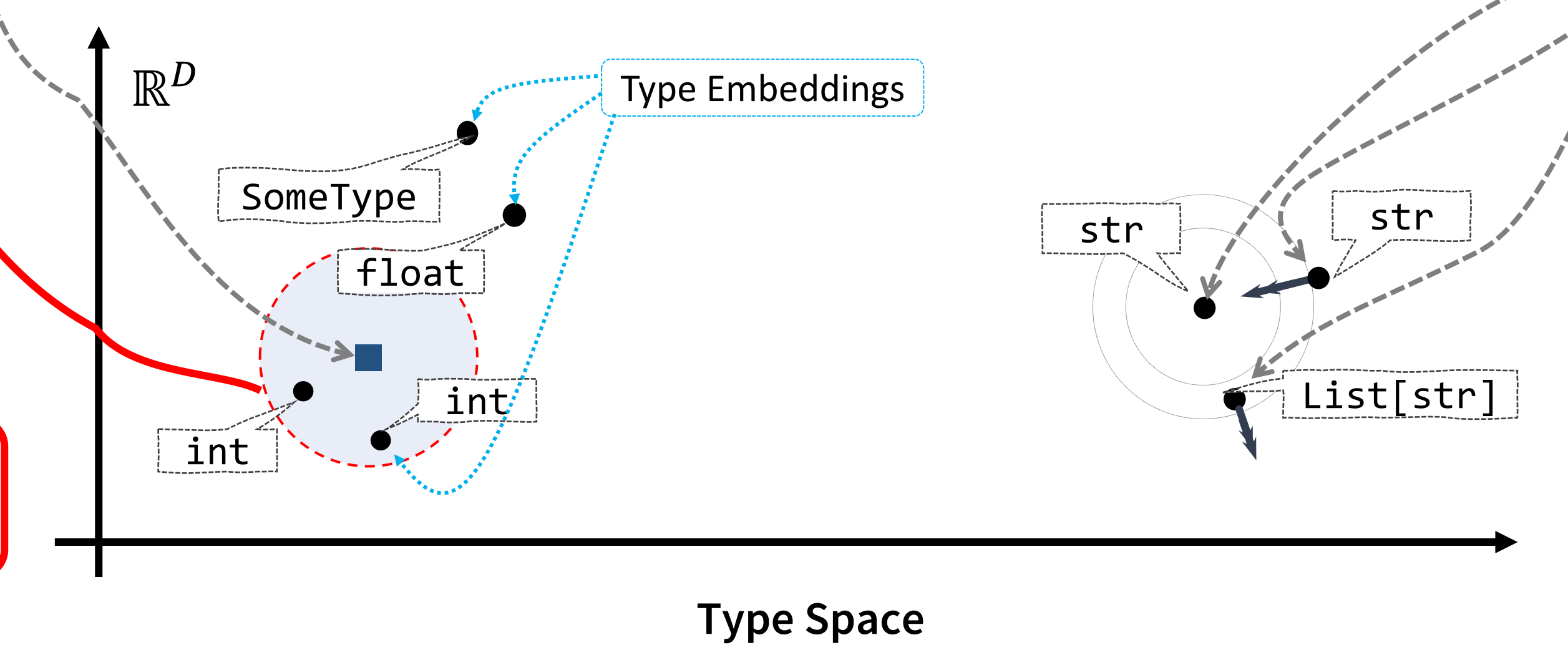
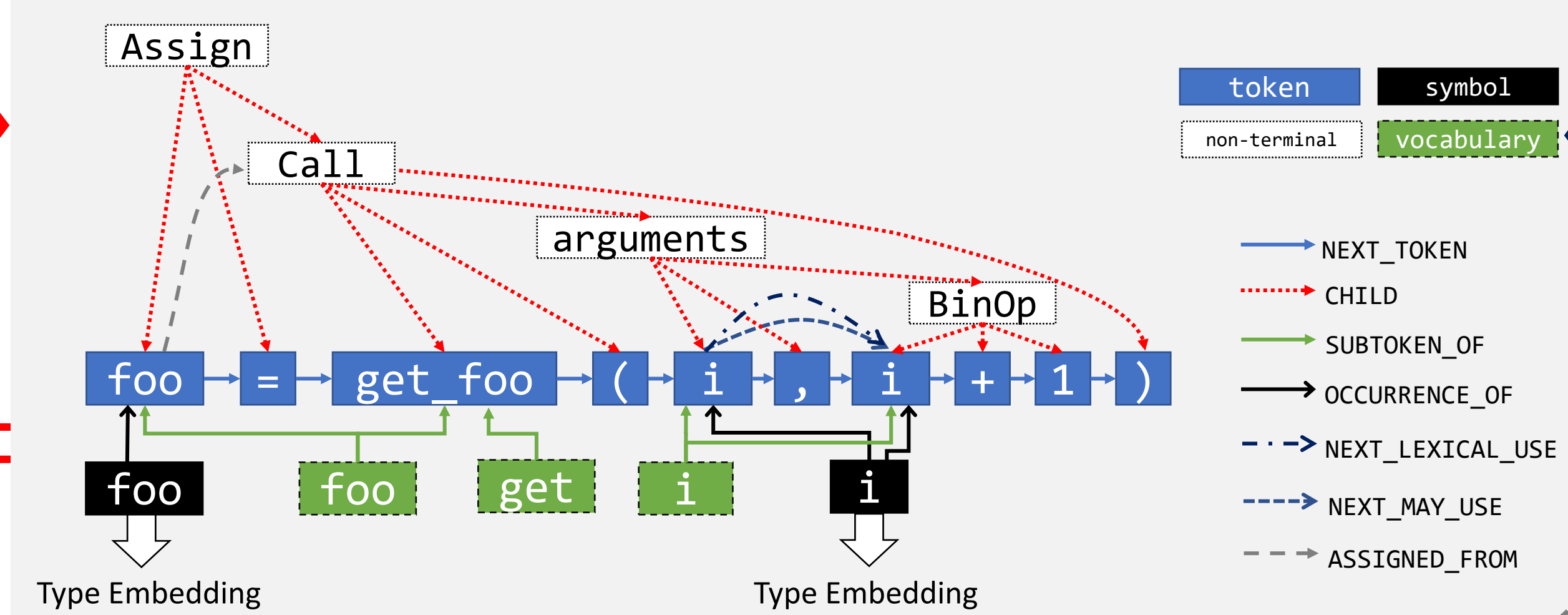


github.com/marketplace/actions/typilus-suggest-python-type-annotations

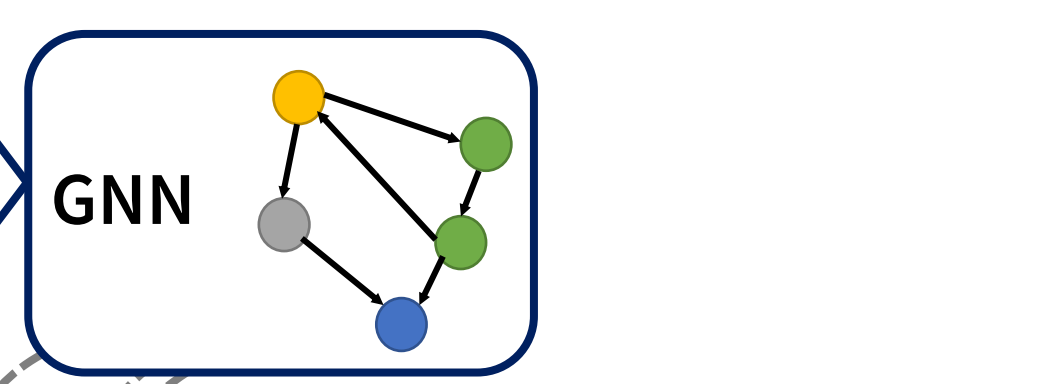
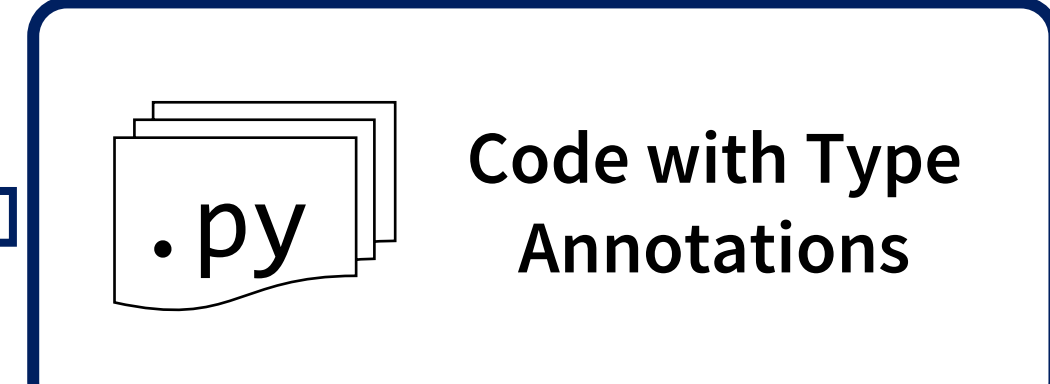
Inference



Graph Representation of Code



Training



Objective

- Deep Similarity Learning
- We use $\mathcal{L}_{\text{space}}$ a variant of triplet loss

$$\max(m, \|r - r_-\| - \|r - r_+\|)$$
- Type embeddings of symbols with the same type (r, r_+) should be close.
- Type embeddings of symbols with different types (r, r_+) should be far away.
- $\mathcal{L}_{\text{Typilus}}$ combines $\mathcal{L}_{\text{space}}$ with a classification loss.

Adaptive Learning

New types can be easily added to the type space:

- Compute type embedding of symbol
- Update type map.
- No re-training needed for the GNN.

Evaluation

How accurately can the models correctly predict ground-truth, developer-provided type annotations?

Types	All	Common	Rare
Seq2Class	39.6%	63.8%	4.6%
Seq2Space	47.4%	62.2%	24.5%
SeqTypilus	52.4%	71.7%	24.9%
Graph2Class	46.1%	74.5%	5.9%
Graph2Space	50.5%	69.7%	23.1%
Typilus	54.6%	77.2%	22.5%

How accurately can models predict type annotations that an optional type checker consider type-correct?

Before	After	mypy	pytype
Annotation	Different Annotation	85%	63%
No Annotation	Annotation	89%	83%