Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心


# 基于图神经网络的二进制程序函数相似性检测

硕士研究生 邢继媛

2021年4月24日

# 内容提要

- 背景简介
- 基本概念
- 算法原理
- 优劣分析
- 应用总结
- 参考文献

# 背景简介

- 预期收获
  - 1.了解二进制程序函数相似性检测任务的基本概念
  - 2.理解基于图神经网络的二进制程序函数相似性检测方法
  - 3.了解二进制程序函数相似性检测的应用

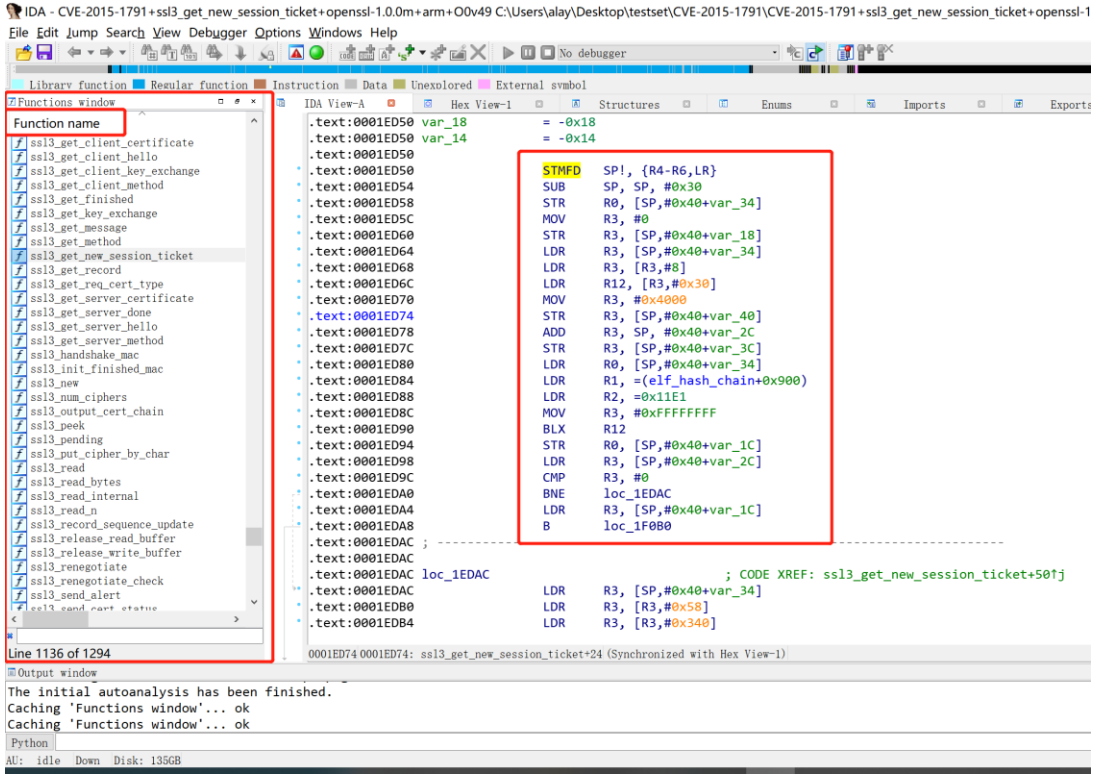
# 基本概念

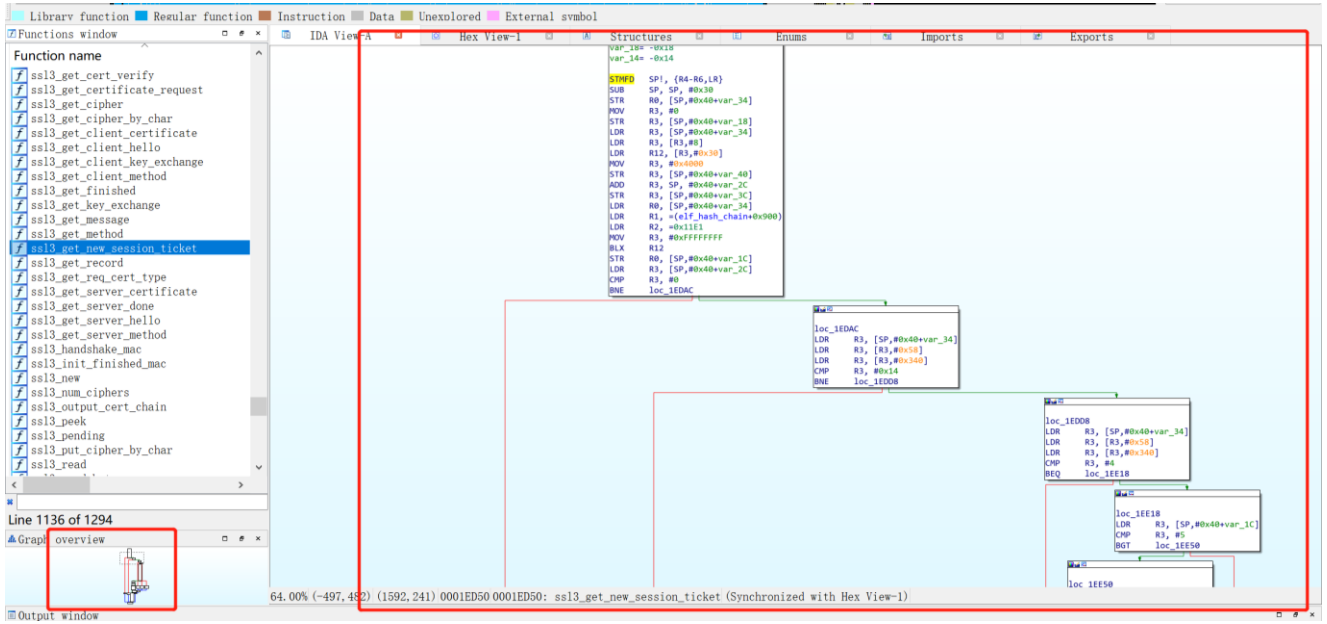
# 基本概念

- 二进制程序函数相似性检测
  - 概念：指检测不同平台，不同编译器，不同优化选项，不同软件版本的两个二进制程序函数的相似程度
    - 平台：X86, ARM, MIPS, POWER PC
    - 编译器：GCC, Clang
    - 优化选项：O0, O1, O2, O3
  - 应用场景：恶意软件分析，版权纠纷，代码抄袭检测，漏洞检测

# 基本概念

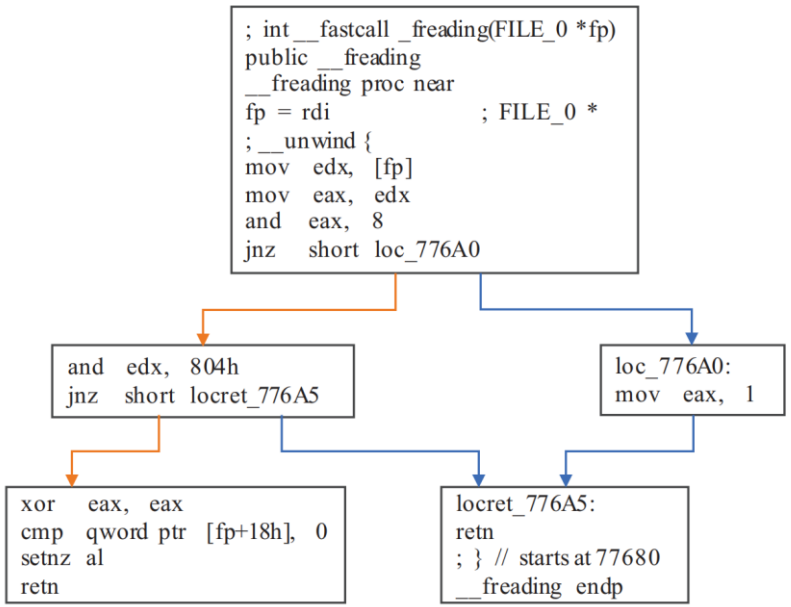
- 二进制程序分析工具：
  - 静态分析工具：IDA Pro、C32Asm
  - 动态调试工具：OD、DEBUG、x64Dbg

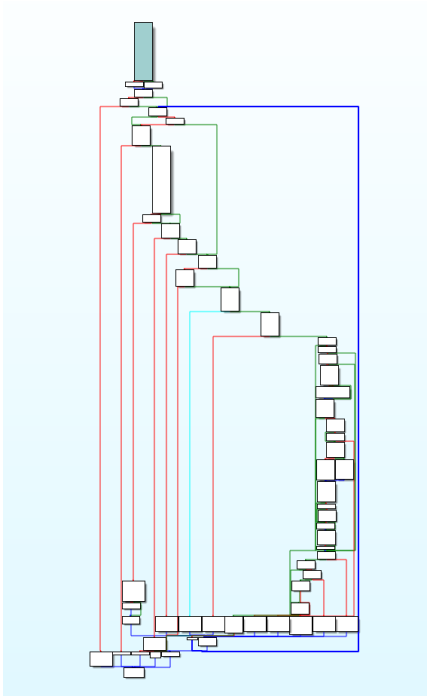



# 基本概念

- 控制流程图（CFG）
  - 有向图 G（V，E）
    - 结点：基本块
    - 边：控制流

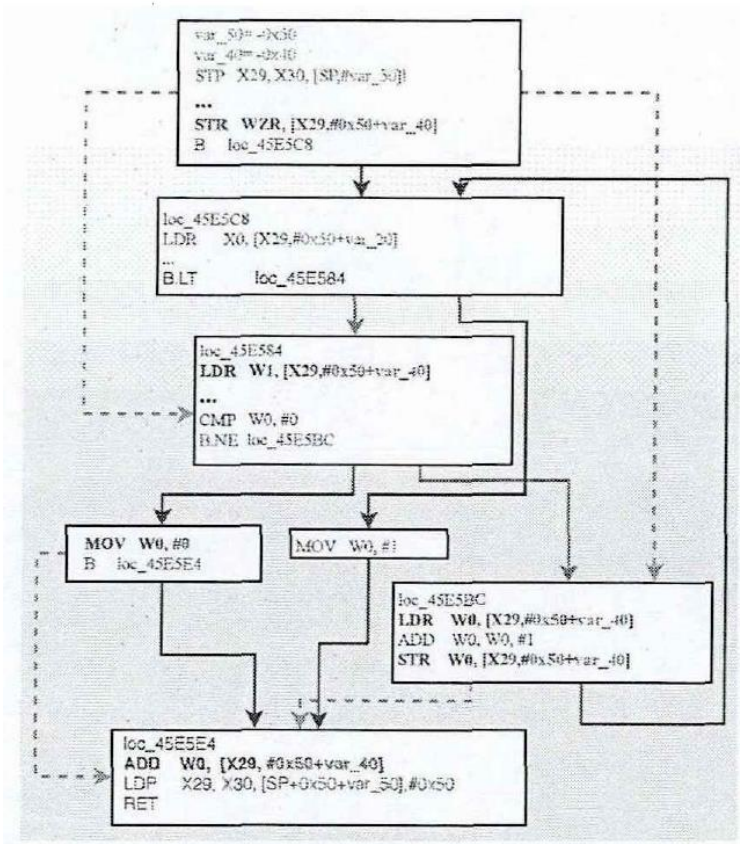


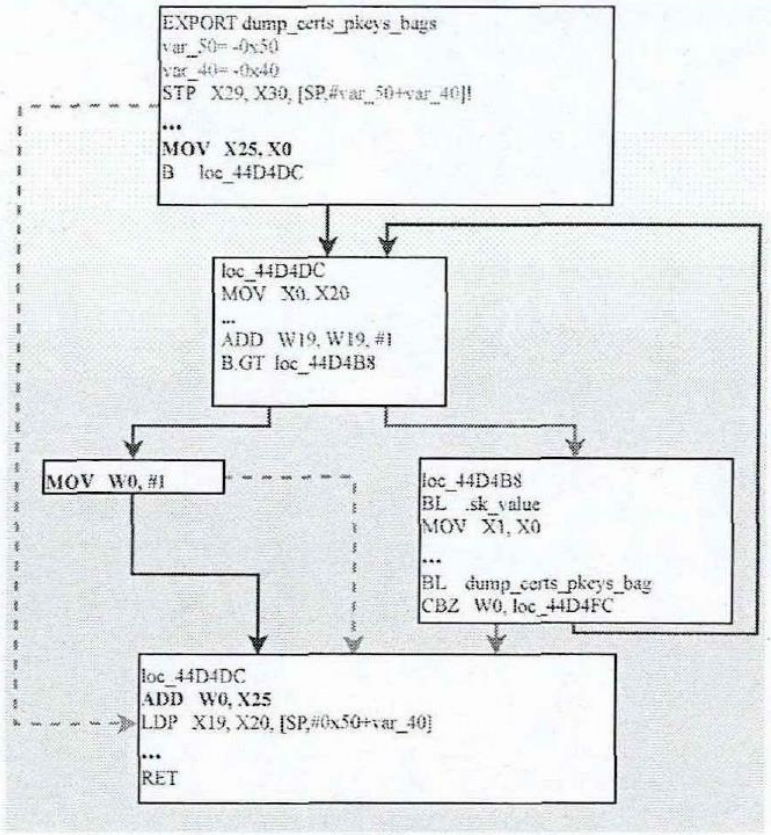
5个基本块

包含不同基本块的函数控制流程图

- 控制流程图（CFG）
  - 同一函数在不同优化选项下的控制流程图

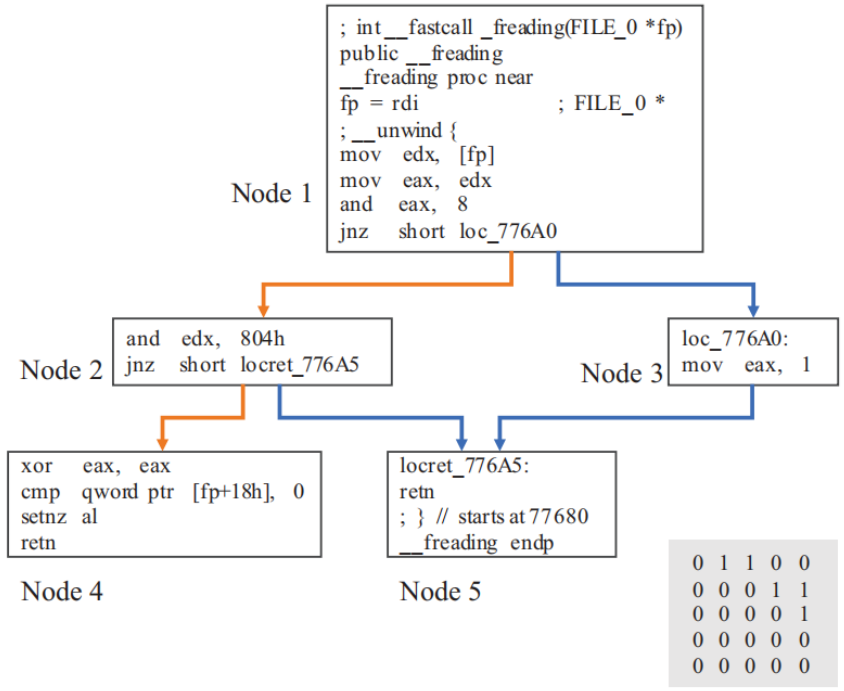



优化选项：O0

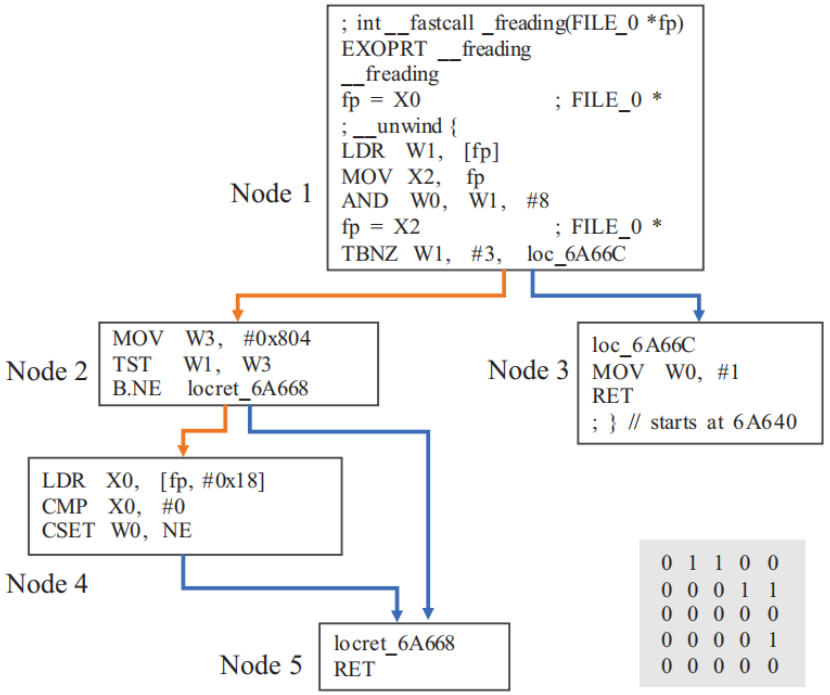



优化选项：O3

- 控制流程图（CFG）
  - 同一函数在不同平台下的控制流程图



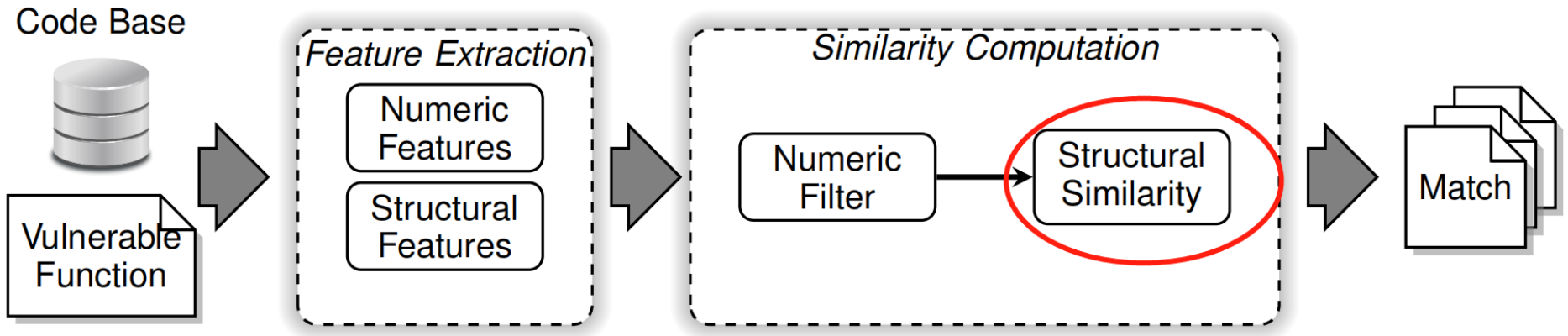

X86-64平台




ARM平台

# 基本概念

- 二进制程序函数相似性检测（基于图匹配）



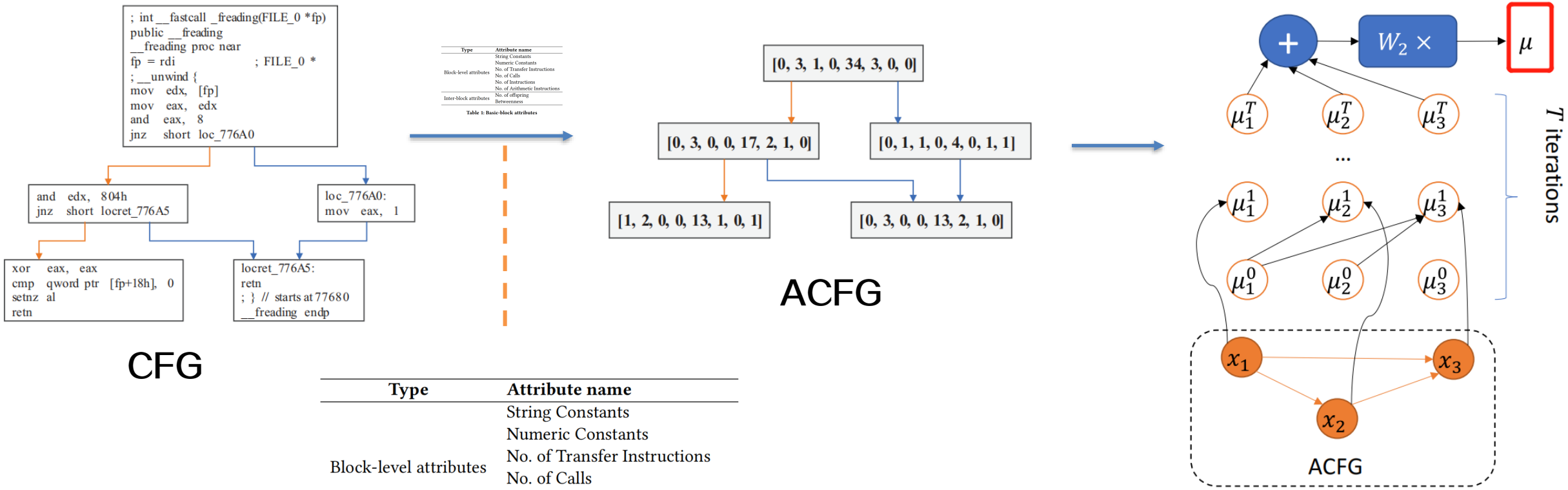

$$d_{mcs.orig}(G_1, G_2) := 1 - \frac{|mcs(G_1, G_2)|}{\max(|G_1|, |G_2|)}$$

$$d_{BB} = \frac{\sum \alpha_i |c_{if} - c_{ig}|}{\sum \alpha_i \max(c_{if}, c_{ig})}$$

$$d_{mcs}(G_1, G_2) := 1 - \frac{|mcs(G_1, G_2)| - \sum d_{BB}(b_i, b_j)}{\max(|G_1|, |G_2|)}$$

# 基本概念

- 二进制程序函数相似性检测（基于图嵌入）




| Type | Attribute name |
|---|---|
| Block-level attributes | String Constants |
| | Numeric Constants |
| | No. of Transfer Instructions |
| | No. of Calls |
| | No. of Instructions |
| | No. of Arithmetic Instructions |
| Inter-block attributes | No. of offspring |
| | Betweenness |

**Table 1: Basic-block attributes**

# 算法原理

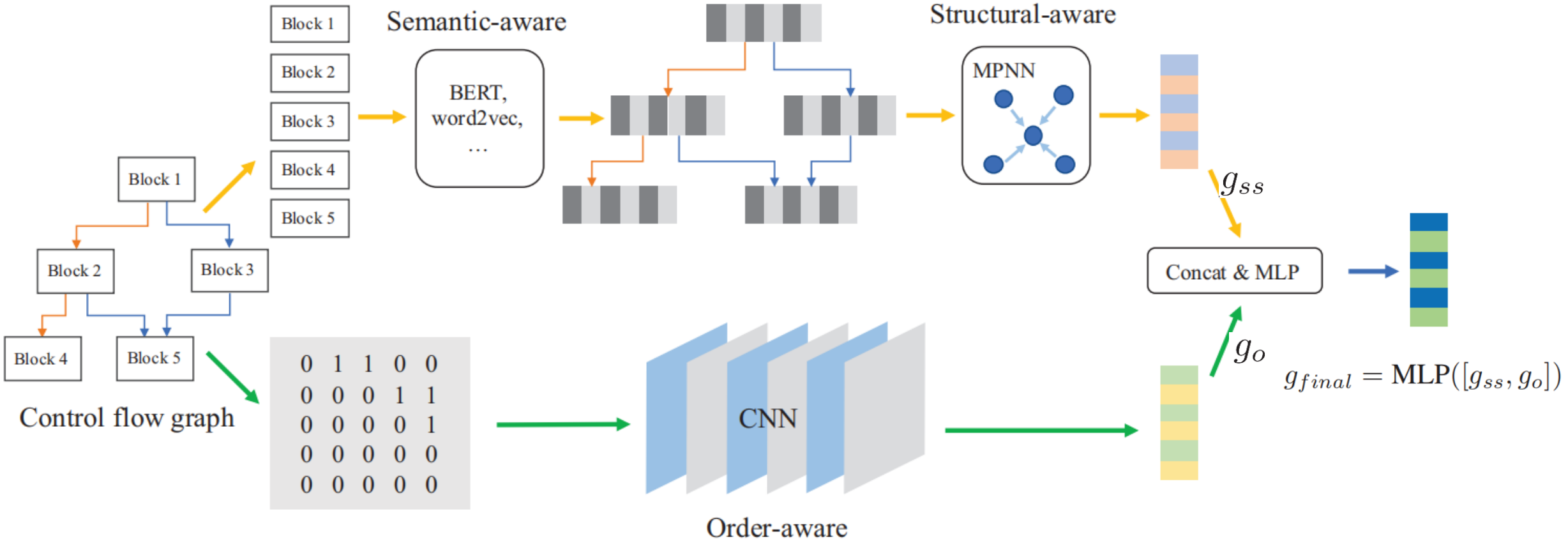
# 算法原理

| | |
|---|---|
| T | 二进制程序函数的相似性检测 |
| I | 二进制程序函数 |
| P | 1. 通过反汇编工具抽取二进制程序函数的CFG<br>2. 通过BERT将CFG中每个基本块嵌入到向量空间，并与CFG结合得到CFG’，通过图神经网络将CFG’嵌入到向量空间，记为$g_{ss}$<br>3. 通过CNN学习CFG中结点顺序特征，表示成向量$g_o$<br>4.拼接向量$[g_{ss}, g_o]$，输入到MLP中，得到二进制程序函数的嵌入向量<br>5.计算两个二进制程序函数嵌入向量的余弦距离作为两个二进制程序函数的相似度 |
| O | 二进制程序函数的相似度 |

| | |
|---|---|
| P | 人工选取基本块特征会损失大量的语义信息 |
| C | 程序可以反汇编 |
| D | 最大化保留CFG中的基本块语义信息和结点顺序信息 |
| L | AAAI-2020 |

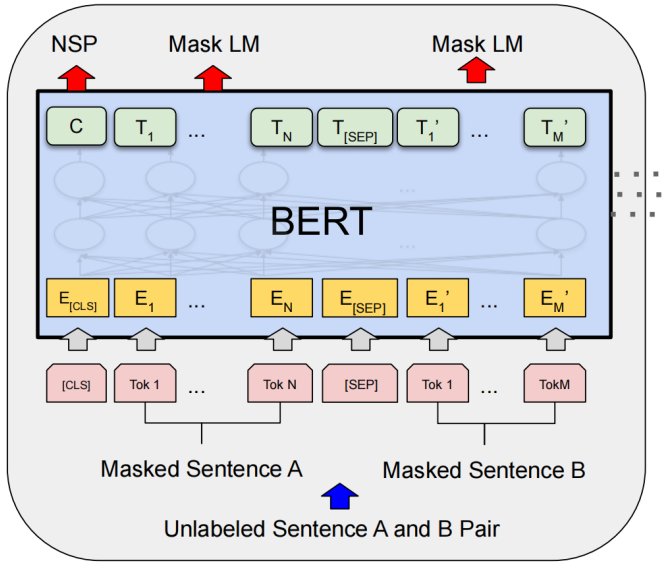
# 算法原理

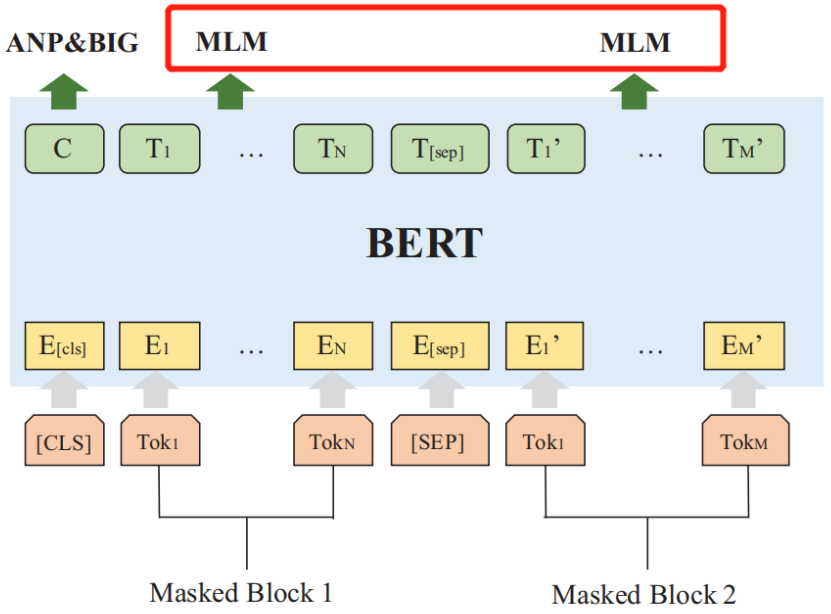
- 基于图神经网络的二进制程序函数相似性检测
  - CFG嵌入流程图



# 算法原理

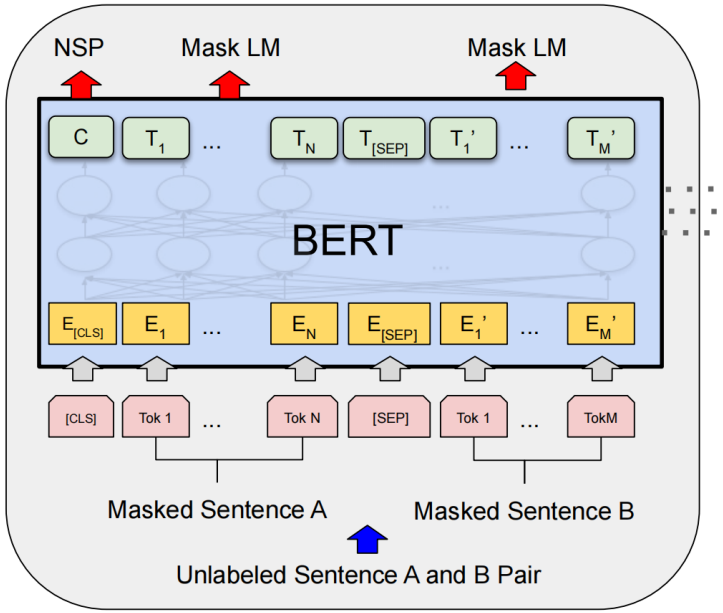
- 基于图神经网络的二进制程序函数相似性检测
  - 语义感知(Semantic aware）
    - 目标：将每个基本块嵌入到向量空间，得到基本块的向量表示
    - 方法：应用BERT模型预训练4个task，学习基本块的语义信息
      - task1：MLM( Masked language model)
        - » 学习基本块内部的语义信息

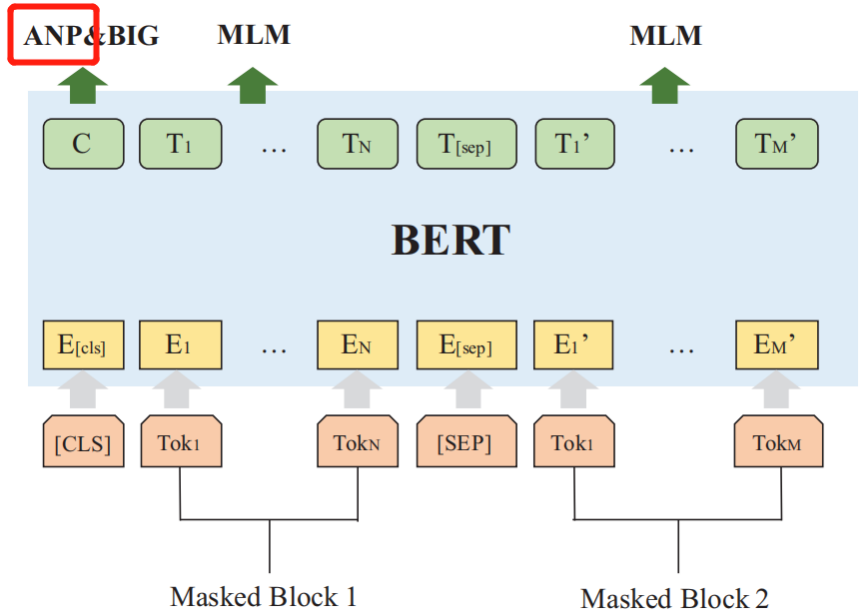


# 算法原理

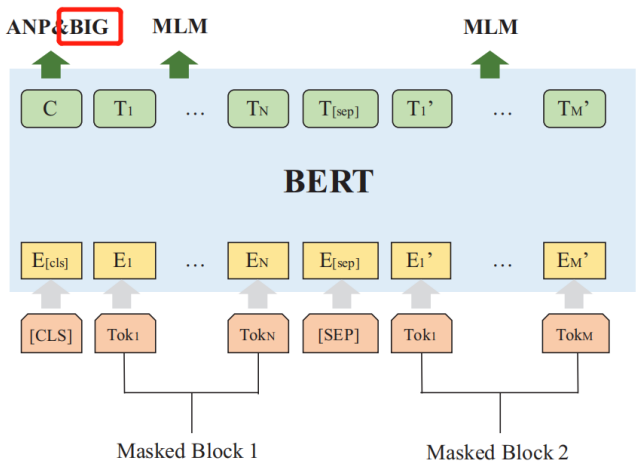
- 基于图神经网络的二进制程序函数相似性检测
  - 语义感知(Semantic aware）
    - 目标：将每个基本块嵌入到向量空间，得到基本块的向量表示
    - 方法：应用BERT模型预训练4个task，学习基本块的语义信息
      - task2 : ANP（Adjacency node prediction）
        - » 判断两个基本块是否临近

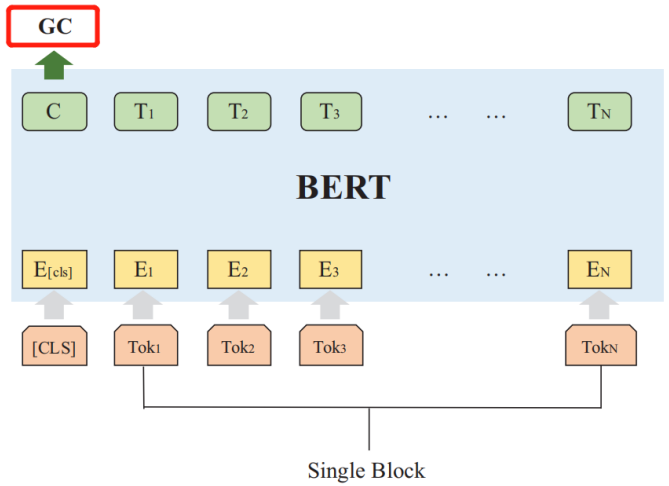


# 算法原理

- 基于图神经网络的二进制程序函数相似性检测
  - 语义感知(Semantic aware）
    - 目标：将每个基本块嵌入到向量空间，得到基本块的向量表示
    - 方法：应用BERT模型预训练4个task，学习基本块的语义信息
      - task3 : BIG（Block inside graph）
        - » 判断两个基本块是否属于同一个图
      - task4 : GC( Graph classification)
        - » 分类基本块所属的平台，编译器，优化选项



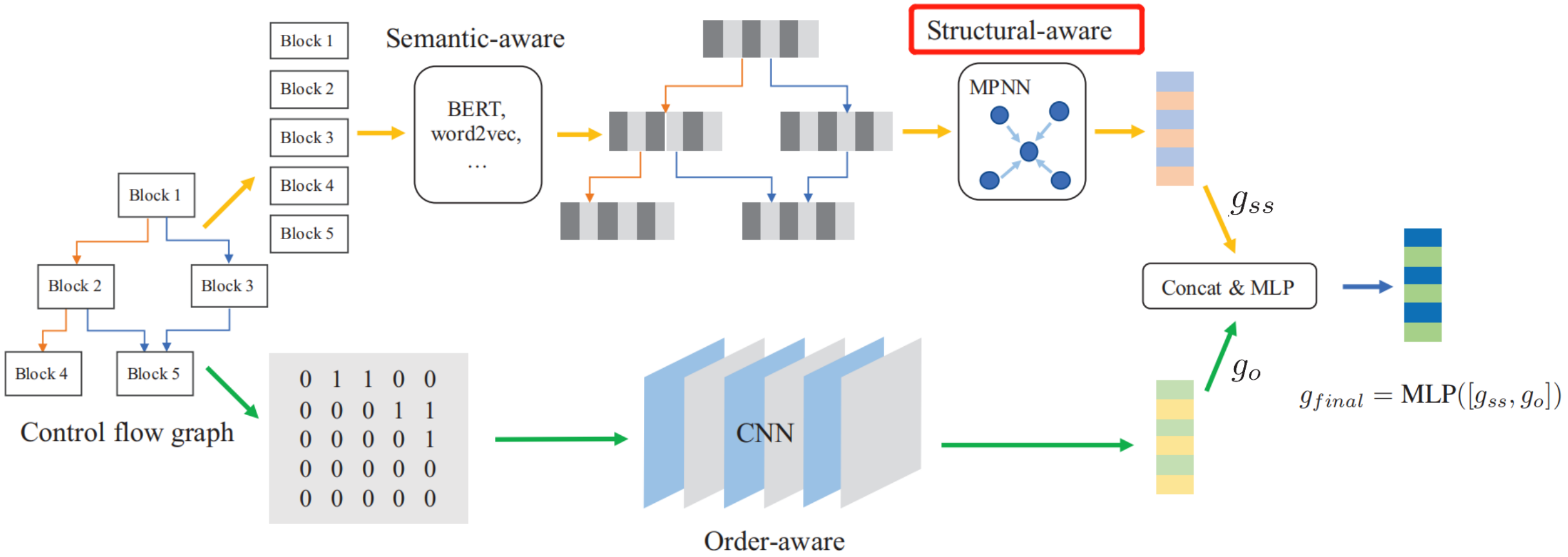
# 算法原理

- 基于图神经网络的二进制程序函数相似性检测
  - 语义感知(Semantic aware）
    - 目标：将每个基本块嵌入到向量空间，得到基本块的向量表示
    - 方法：应用BERT模型预训练4个task，学习基本块的语义信息

| | task | T | level |
|---|---|---|---|
| 1 | MLM | 学习基本块内部的语义信息 | token-level |
| 2 | ANP | 判断两个基本块是否临近 | block-level |
| 3 | BIG | 判断两个基本块是否属于同一个图 | graph-level |
| 4 | GC | 分类基本块所属的平台，编译器，优化选项 | graph-level |

# 算法原理

- 基于图神经网络的二进制程序函数相似性检测
  - CFG嵌入流程图



# 算法原理

- 基于图神经网络的二进制程序函数相似性检测
  - 结构感知(Structural aware)
    - 目标：将CFG’嵌入到向量空间，得到CFG’的向量表示
    - 方法：使用MPNN（信息传递网络——图神经网络通用框架）
      - MPNN：信息传递阶段+读取阶段

» 信息传递阶段（message passing）
- 迭代更新结点的向量表示
- U： 结点更新函数
- M：消息函数

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

» 读取阶段（readout）
- 通过读取函数R得到整张图的向量表示
- R：读取函数

$$g_{ss} = R(h_v^T | v \in G)$$

# 算法原理

- 基于图神经网络的二进制程序函数相似性检测
  - 结构感知(Structural aware)
    - 目标：将CFG’嵌入到向量空间，得到CFG’的向量表示
    - 方法：使用MPNN（信息传递网络——图神经网络通用框架）
      - MPNN：信息传递阶段+读取阶段

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

$$g_{ss} = R(h_v^T | v \in G)$$

M：MLP（多层感知器）

U：GRU（循环神经网络）

R：$\sum$

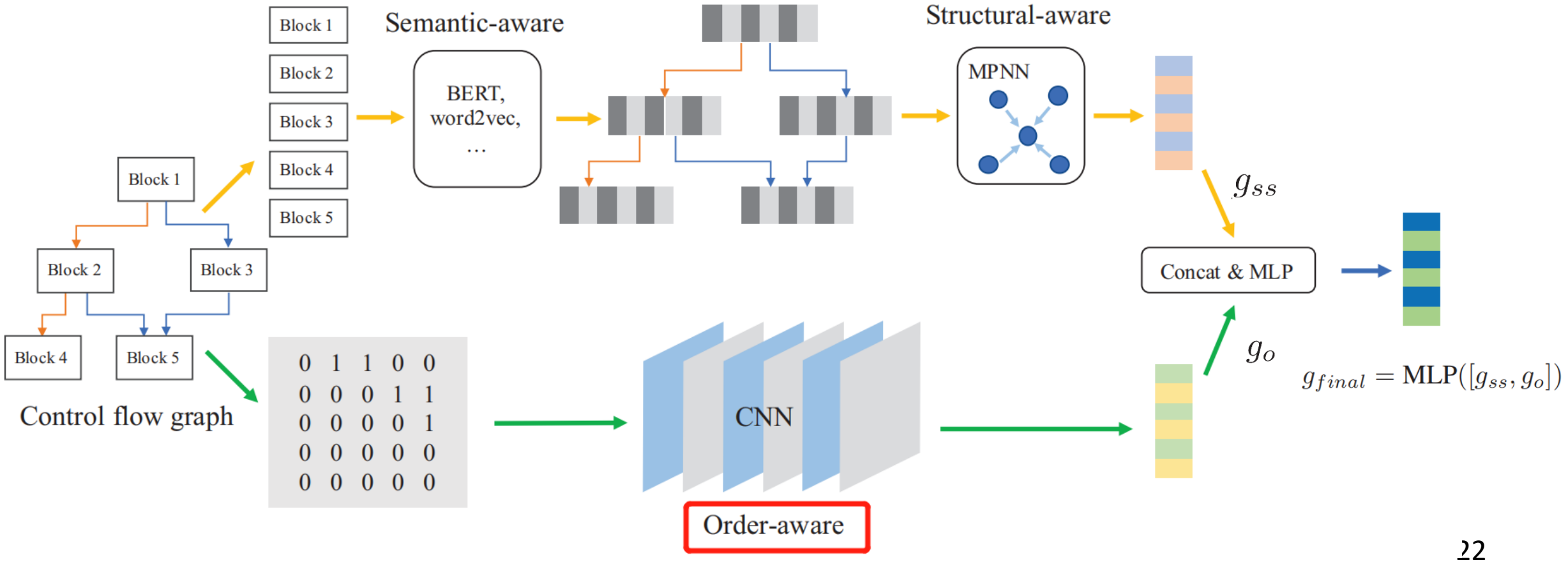
$$m_v^{t+1} = \sum_{w \in N(v)} \mathrm{MLP}(h_w^t)$$

$$h_v^{t+1} = \mathrm{GRU}(h_v^t, m_v^{t+1})$$

$$g_{ss} = \sum_{v \in G} \mathrm{MLP}(h_v^0, h_v^T)$$

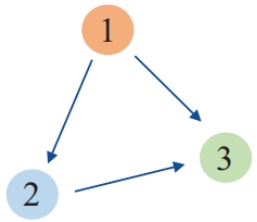
# 算法原理

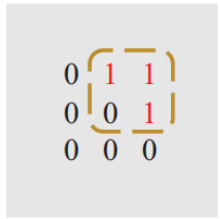
- 基于图神经网络的二进制程序函数相似性检测
  - CFG嵌入流程图

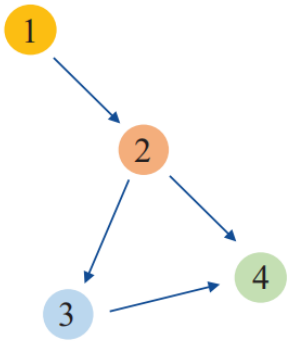


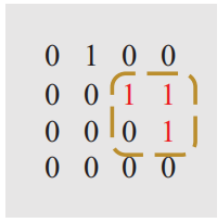
# 算法原理

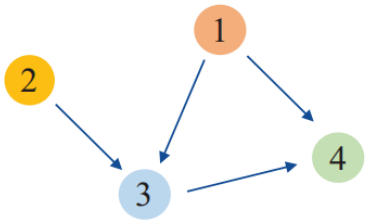
- 基于图神经网络的二进制程序函数相似性检测
  - 结点顺序感知(Node order aware)
    - 目标：学习CFG中结点顺序特征
    - 方法：通过CNN学习邻接矩阵的特征，生成向量$g_o$

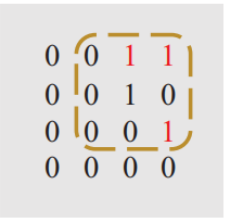


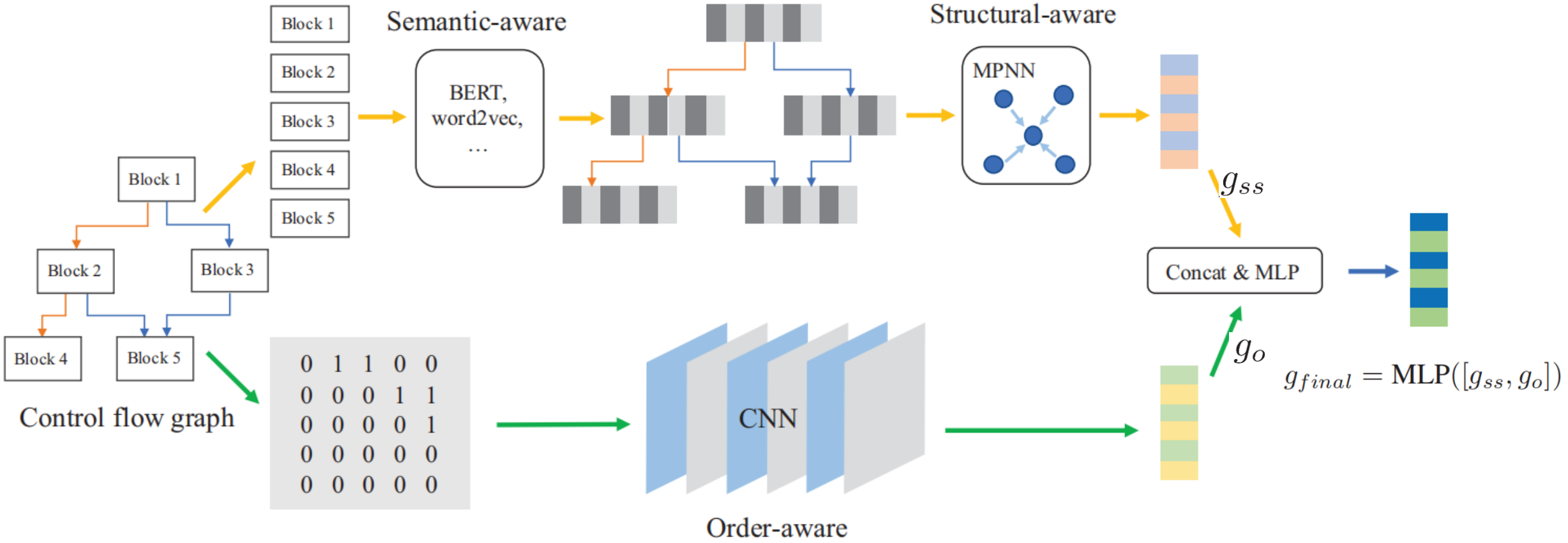
# 算法原理

- 基于图神经网络的二进制程序函数相似性检测
  - 结点顺序感知(Node order aware)
    - 目标：学习CFG中结点顺序特征
    - 方法：通过CNN学习邻接矩阵的特征，生成向量$g_o$
    - 优点：
      - CNN的平移不变性和尺度不变性
      - 输入可以为不同结点数目的图
    - 计算：

$$g_o = \text{Maxpooling}(\text{Resnet}(A))$$

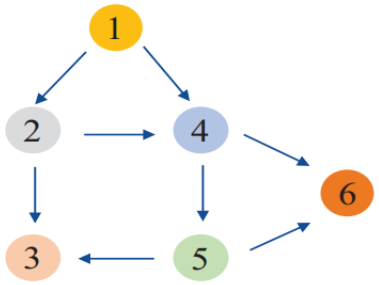
# 算法原理

- 基于图神经网络的二进制程序函数相似性检测
  - CFG嵌入流程图



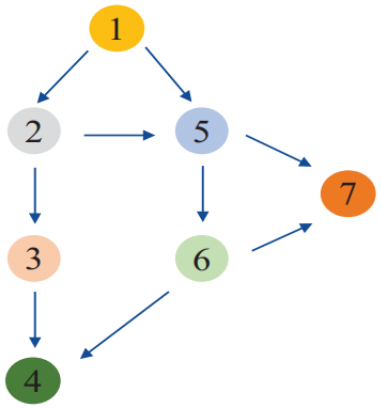
# 算法原理

- 基于图神经网络的二进制程序函数相似性检测
  - 实验结果举例
    - Cosine = 0.971




| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 |

X86-64




| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |

ARM

# 算法原理

- 基于图神经网络的二进制程序函数相似性检测
  - 实验结果
    - 评价指标
      - Rank-n
        - » query的前n个结果中有正确结果的query数占总query数的比例
      - MRR-n
        - » 对于一个query，搜索结果中第一个正确答案在前n个结果中排在第$r_i$位，则得分为$\frac{1}{r_i}$，对所有query得分取平均
        - » $\mathrm{MMR} = \frac{1}{Q}\sum_{i=1}^{Q}\frac{1}{r_i}$

| QUERY | Results | Rank-3 | MRR-3 |
|---|---|---|---|
| Taylor | TS,Taylor,lover,sq | $\frac{2}{3}$ | $\frac{1}{2} = \frac{1}{3}(\frac{1}{2} + \frac{1}{1} + 0)$ |
| Miao | Miao,cat,dog,bobo | | |
| LAY | Slay,zyx,cool,LAY | | |

# 算法原理

- 基于图神经网络的二进制程序函数相似性检测
  - 实验1——相似性检测

| Dataset | Task | Training | Validation | Testing |
|---|---|---|---|---|
| gcc-O2 | 1 | 31,410 | 3,857 | 3,884 |
| gcc-O3 | 1 | 16,059 | 4,155 | 4,077 |
| gcc-x86-64 | 2 | 27,761 | 3,406 | 3,492 |
| gcc-ARM | 2 | 9,447 | 4,773 | 4,933 |

1. 更新函数GRU： 在结点更新时可以存储更多的信息
2. 基本块嵌入：采用NLP方法提取特征优于人工提取特征
3. 语义信息比结点顺序信息更有效

方法效果对比

| Model | Task1-O2 | Task1-O3 |
|---|---|---|
| Weisfeiler-Lehman | 0.2493 / 0.1810 | 0.1940 / 0.1565 |
| Gemini | 0.6069 / 0.5491 | 0.5430 / 0.4760 |
| MPNN | 0.6096 / 0.5507 | 0.5501 / 0.4802 |
| word2vec | 0.7003 / 0.6534 | 0.6198 / 0.5555 |
| skip thought | 0.6825 / 0.6238 | 0.5954 / 0.5226 |
| BERT2 | 0.7591 / 0.7060 | 0.6507 / 0.5852 |
| BERT4 | 0.7704 / 0.7233 | 0.6672 / 0.5989 |
| CNN3 (random) | 0.0362 / 0.0020 | 0.0307 / 0.0015 |
| CNN3 | 0.4142 / 0.3684 | 0.3074 / 0.2612 |
| Resnet7 | 0.4330 / 0.3868 | 0.3229 / 0.2732 |
| Resnet11 | 0.4419 / 0.3952 | 0.3271 / 0.2837 |
| $\text{MPNN}_{ws}$ | 0.3361 / 0.3014 | 0.2161 / 0.1913 |
| $\text{MPNN}_{ws}$+Resnet11 | 0.4457 / 0.3970 | 0.3348 / 0.2907 |
| Our model | **0.7922 / 0.7421** | **0.6855 / 0.6114** |

结构感知对比

语义感知对比

结点顺序感知对比

MRR10 / Rank1

# 算法原理

- 基于图神经网络的二进制程序函数相似性检测
  - 实验1——相似性检测

1. 双向Transformer可以提取基本块中更多语义信息
2. Graph-level信息的有效性

1. CNN可以学习到结点顺序信息
2. 残差网络效果优于CNN
3. 结点顺序信息的有效性

| Model | Task1-O2 | Task1-O3 |
|---|---|---|
| Weisfeiler-Lehman | 0.2493 / 0.1810 | 0.1940 / 0.1565 |
| Gemini | 0.6069 / 0.5491 | 0.5430 / 0.4760 |
| MPNN | 0.6096 / 0.5507 | 0.5501 / 0.4802 |
| word2vec | 0.7003 / 0.6534 | 0.6198 / 0.5555 |
| skip thought | 0.6825 / 0.6238 | 0.5954 / 0.5226 |
| BERT2 | 0.7591 / 0.7060 | 0.6507 / 0.5852 |
| BERT4 | 0.7704 / 0.7233 | 0.6672 / 0.5989 |
| CNN3 (random) | 0.0362 / 0.0020 | 0.0307 / 0.0015 |
| CNN3 | 0.4142 / 0.3684 | 0.3074 / 0.2612 |
| Resnet7 | 0.4330 / 0.3868 | 0.3229 / 0.2732 |
| Resnet11 | 0.4419 / 0.3952 | 0.3271 / 0.2837 |
| $\text{MPNN}_{ws}$ | 0.3361 / 0.3014 | 0.2161 / 0.1913 |
| $\text{MPNN}_{ws}$+Resnet11 | 0.4457 / 0.3970 | 0.3348 / 0.2907 |
| Our model | **0.7922 / 0.7421** | **0.6855 / 0.6114** |

方法效果对比

结构感知对比

语义感知对比

结点顺序感知对比

MRR10 / Rank1

# 算法原理

- 基于图神经网络的二进制程序函数相似性检测
  - 实验2——区分不同优化选项

| Dataset | Task | Training | Validation | Testing |
|---|---|---|---|---|
| gcc-O2 | 1 | 31,410 | 3,857 | 3,884 |
| gcc-O3 | 1 | 16,059 | 4,155 | 4,077 |
| gcc-x86-64 | 2 | 27,761 | 3,406 | 3,492 |
| gcc-ARM | 2 | 9,447 | 4,773 | 4,933 |

1. 可以有效区分不同的优化选项
2. 可以区分不同的编译器，但是作者没有在相关数据集上进行实验验证

| Model | Task2-x86-64 | Task2-ARM |
|---|---|---|
| Weisfeiler-Lehman | - | - |
| Gemini | 77.88 | 79.89 |
| MPNN | 79.65 | 80.62 |
| word2vec | 82.24 | 84.23 |
| skip thought | 80.43 | 83.74 |
| BERT2 | 82.67 | 85.19 |
| BERT4 | 83.74 | 86.33 |
| CNN3 (random) | 66.06 | 64.57 |
| CNN3 | 82.11 | 83.70 |
| Resnet7 | 82.56 | 84.13 |
| Resnet11 | 82.64 | 84.24 |
| $\text{MPNN}_{ws}$ | 76.29 | 76.90 |
| $\text{MPNN}_{ws}$+Resnet11 | 82.92 | 85.05 |
| Our model | **86.14** | **88.41** |

准确率

# 优劣分析

- 横向对比
  - 图匹配：大规模检测时间效率低；迁移应用困难
  - 图嵌入：大规模检测时间效率高；迁移应用容易
    - 迁移应用
      - 代码抄袭检测场景
      - 漏洞检测场景
- 纵向对比
  - 保留了CFG中的基本块语义信息和结点顺序信息

# 应用总结

- 恶意软件分析
- 版权纠纷
- 代码抄袭检测
- 漏洞搜索
- 同源漏洞判别

# 参考文献

[1] Yu Z, Cao R, Tang Q, et al. Order matters: semantic-aware neural networks for binary code similarity detection[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(01): 1145-1152.
[2] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018. (BERT)
[3] Gilmer J, Schoenholz S S, Riley P F, et al. Neural message passing for quantum chemistry[C]//International Conference on Machine Learning. PMLR, 2017: 1263-1272. (MPNN)

上善若水。水善利万物
而不争，处众人之所恶，
故几於道。居善地，心
善渊与善仁，言善信，
正善治，事善能，动善
时。夫唯不争，故无尤。

# 谢谢！