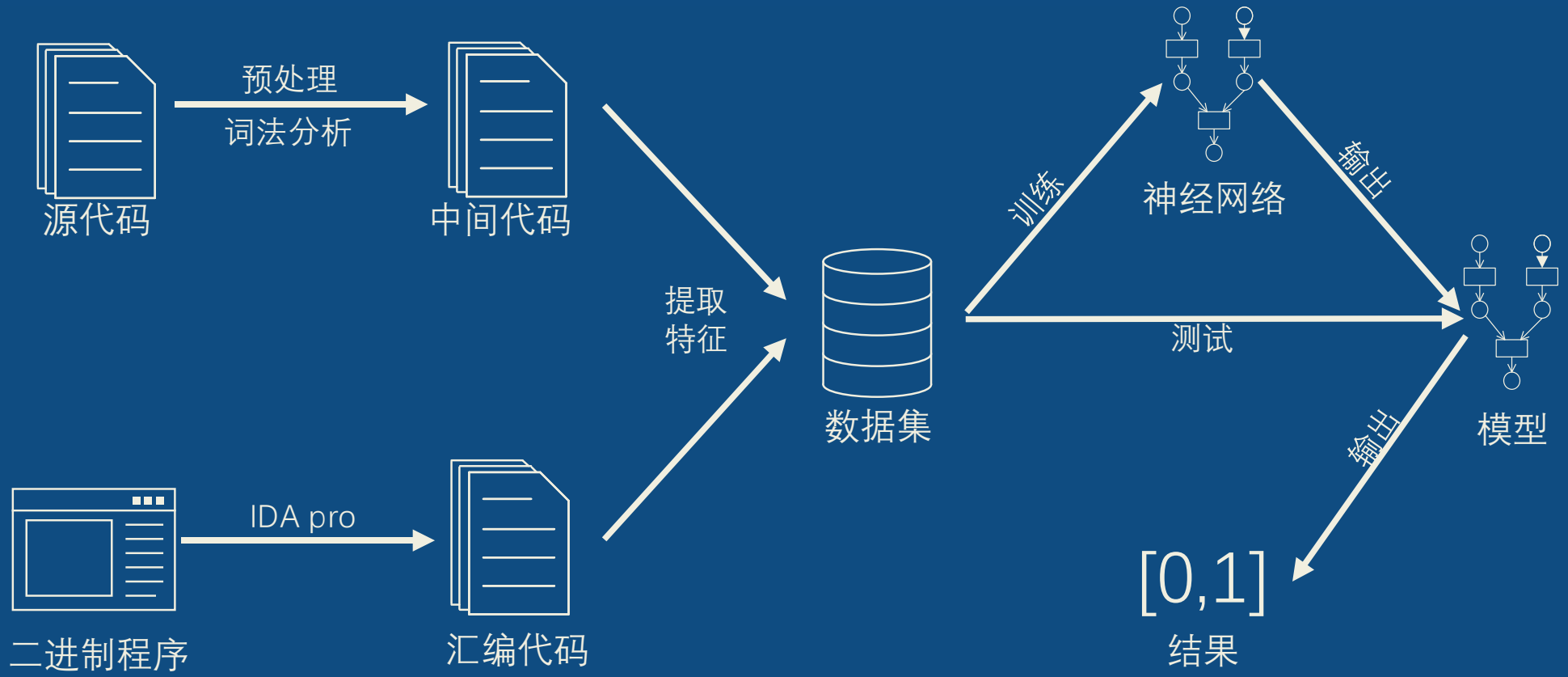
華中科技大學
网络空间安全学院
School of Cyber Science and Engineering, HUST

框架设计

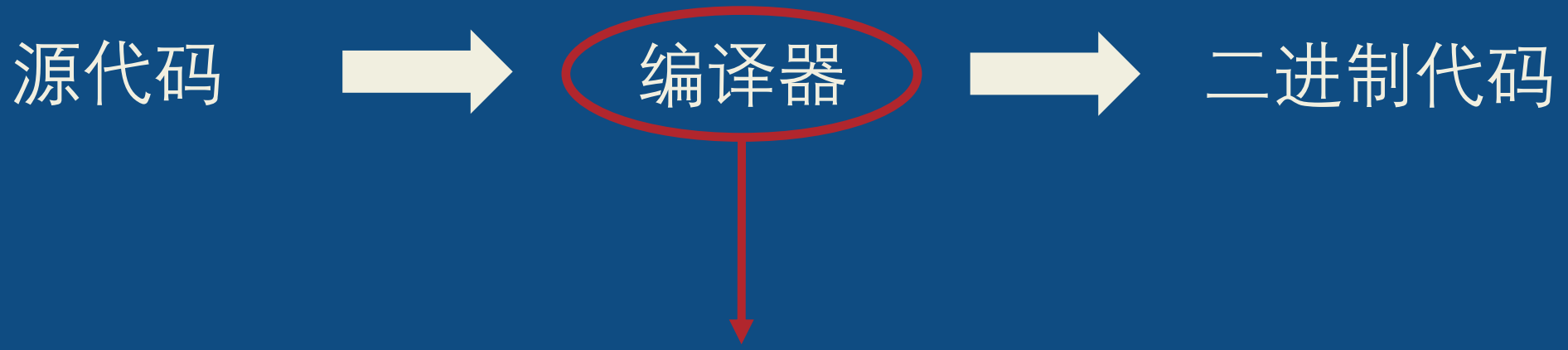
跨类代码 函数级 相似性检测



華中科技大學
网络空间安全学院
School of Cyber Science and Engineering, HUST

特征选择与提取


基本块特征
融合特征流
LCFG
LFG
CFG



CFG

源代码与二进制代码都可以分析得到其对应的CFG

源代码与其编译得到的二进制代码的CFG之间具有强相关性


基本块特征
融合特征流
LCFG
LFG
CFG

基本块特征

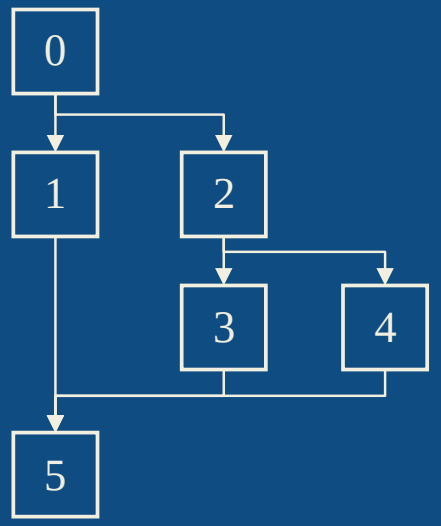
代码字面量

C语言	二进制语言
代码行数	指令数
字符串数量	字符串数量
数字数量	立即数数量
计算指令数量	计算指令数量
位移指令数量	位移指令数量
函数调用数量	函数调用指令数量
返回命令数量	返回指令数量

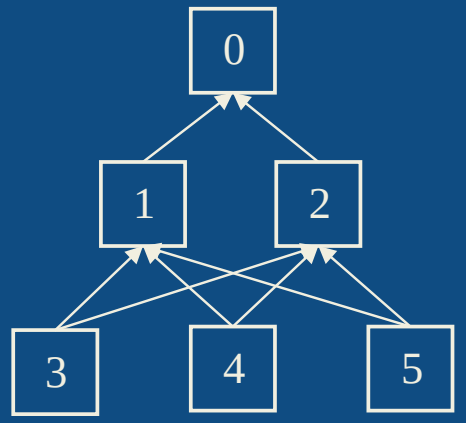
代码语义序列

指令序列	关键字序列
add	+=
call	=
mov	(
cmp	)
jz	==

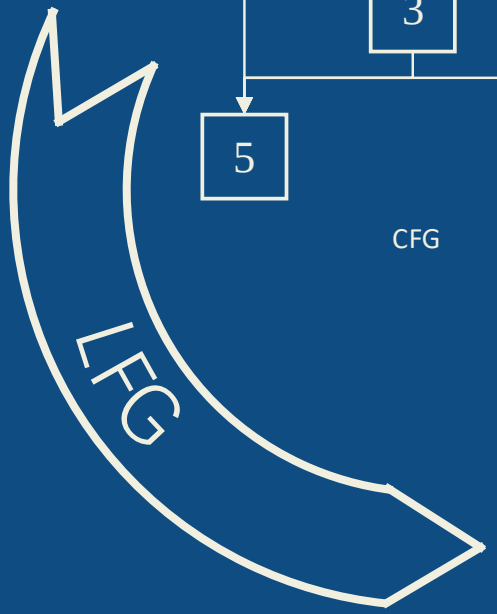

基本块特征
融合特征流
LCFG
LFG
CFG



CFG



LFG



LFG提取方法

1) 提取目标函数的CFG
2) 以函数入口基本块为根节点，依照CFG对其它基本块进行广度优先搜索确定基本块层级
3) 按照基本块之间的层级关系生成基本块间路径

LFG优势

1) LFG可以直接从CFG中提取
2) LFG可以使CFG上“距离”较远的点“拉近”
3) LFG的全连接结构很适合结合MLP的思想
4) LFG可以削减分支中过长旁路的影响


基本块特征
融合特征流
LCFG
LFG
CFG

C语言代码特征提取


```
{
    "Funcname":   "ASN1_d2i_bio",
    "Nodenum":    4,
    "CFG":        [[1, 2], [3], [3], []],
    "Literal":    [[6, 0, 0, 6, 6, 2, 0],
                   [1, 0, 0, 0, 0, 0, 0],
                   [2, 0, 0, 1, 1, 2, 0],
                   [3, 0, 0, 0, 0, 1, 1]],
    "Semantic":   [[7, 33, 38, 38, 7, 39, 39, 7, 7, 33, 38, 38, 7, 39, 39, 33, 40, 38, 8, 39, 36, 38, 15, 39],
                   [36],
                   [33, 38, 7, 39, 33, 40, 38, 8, 39],
                   [40, 38, 39, 37]]
}
```

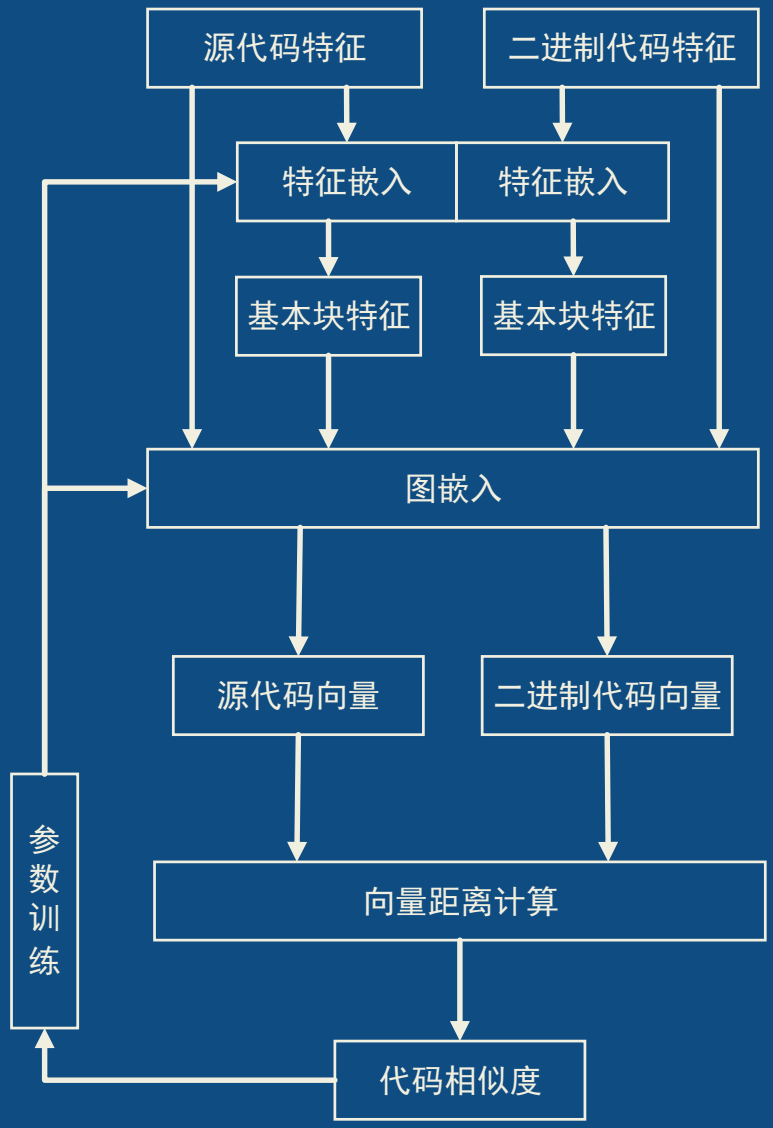
二进制代码特征提取


```
{
    "Funcname":   "ASN1_d2i_bio",
    "Nodenum":    4,
    "CFG":        [[1, 2], [3], [3], []],
    "Literal":    [[17, 4, 0, 2, 0, 1, 0],
                   [12, 0, 0, 0, 0, 1, 0],
                   [1, 0, 0, 0, 0, 0, 0],
                   [6, 0, 0, 0, 0, 1, 1]],
    "Semantic":   [[179, 152, 252, 152, 152, 152, 152, 152, 152, 127, 152, 152, 152, 20, 152, 60, 122],
                   [152, 152, 152, 152, 160, 127, 152, 152, 152, 20, 152, 102],
                   [164],
                   [152, 152, 20, 152, 128, 191]]
}
```

華中科技大學
网络空间安全学院
School of Cyber Science and Engineering, HUST

模型构建与实现

GSN模型整体框架



特征嵌入层

其作用是标准化基本块特征，同时学习不同类型语言之间基本块特征的关联性

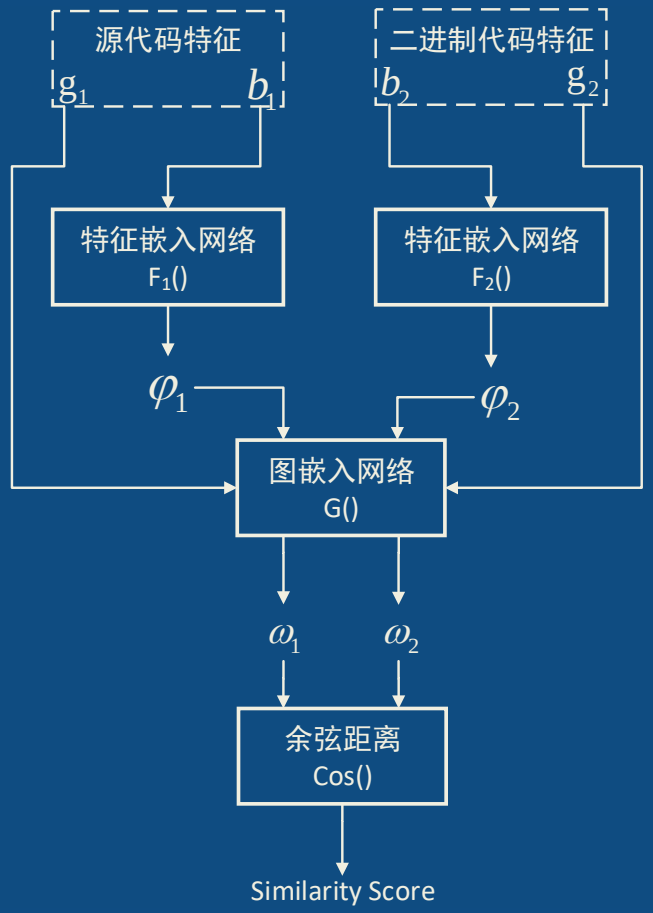
图嵌入层

其作用是结合图特征和基本块特征，生成代表函数的嵌入向量特征

识别层

其作用是判断两种语言函数向量之间的相似度，并在训练时通过反向传播调整前两层中神经网络的参数

伴孪生网络判断框架



$$similarity\ score = \cos(G(g_1, F_1(b_1)), G(g_2, F_2(b_2)))$$

孪生网络: 参数完全共享

伪孪生网络: 参数不共享

伴孪生网络: 参数部分共享

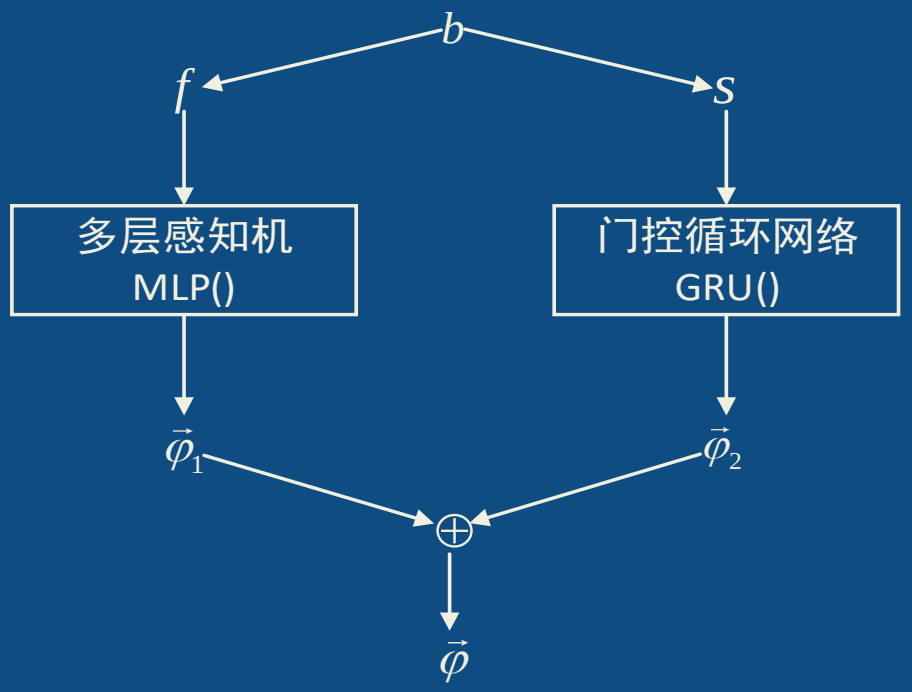
基本块特征嵌入——不共享

不同的语言基本块特征存在明显差别

节点迭代过程——共享

不同语言的大部分CFG可以通过固定变换相互转换

基本块特征处理



$$F(b) = MLP(f) + GRU(s)$$

F: 基本块特征嵌入函数

b: 基本块特征

f: 代码字面量特征

s: 代码序列语义

将基本块特征单向转化为向量特征，学习特征间的内在联系，与欠完备自编码器类似

MLP——结构简单，收敛迅速

GRU——适合序列，收敛迅速

图嵌入过程设计

Structure2vec节点迭代过程

$$\mu_v^{(t+1)} = F(x_v, \sum_{u \in N(v)} \mu_u^{(t)}), \forall v \in V \quad (4\text{-}1)$$

GSN中非线性函数F的设计

$$\mu_v^{(t+1)} = \tanh\left(x_v + MLP\left(\sum_{t \in N(v)} \alpha_{uv}^{(t)} \mu_u^{(t)} \right) \right), \forall v \in V \quad (4\text{-}2)$$

注意力机制的计算过程

$$\alpha_{vu}^{(t)} = \frac{\exp\left(leaky\_relu\left(\vec{a}^T \left[\mu_v^{(t)} \| \mu_u^{(t)} \right]\right)\right)}{\sum_{k \in N(v)} \exp\left(leaky\_relu\left(\vec{a}^T \left[\mu_v^{(t)} \| \mu_k^{(t)} \right]\right)\right)} \quad (4\text{-}3)$$

非线性化计算过程

$$MLP(\mu_v^{(t+0.5)}) = \underbrace{W_n \times relu(W_{n-1} \times \cdots \times relu(W_1 \mu_v^{(t+0.5)})}_{n level} \quad (4\text{-}4)$$

GSN一次节点迭代过程计算公式

$$G(x, CFG, LFG) = w\left(\sum_{v \in V} F(x_v, \sum_{u \in CFG(v)} \mu_u^{(t)}) + \sum_{v \in V} F(x_v, \sum_{u \in LFG(v)} \mu_u^{(t)}) \right) + b \quad (4\text{-}5)$$

$\mu_v^{(t)}$：经过t次迭代后节点v的嵌入，$\mu_v^{(0)}=0$

F：一个通用的非线性函数

X_v：节点的特征向量

N：所有节点的集合

$N(v)$：与节点v*相连*的节点的集合

添加注意力机制

多个图嵌入融合

GSN模型分析

引入注意力机制的节点迭代

GSN通过注意力机制带来的权重，也可以隐含地体现CFG在执行时的主要路径，这也为模型的可解释性带来了优势。

使用双向传播代替传统GAT网络中的单向传播，反向传播信息可以作为正向传播信息的一种良好补充。

引入注意力机制，解决了优化带来CFG差别的情况，使在不同CFG结构上计算得到相似的图嵌入向量得以实现。

基于跨模态检索的伴孪生网络

GSN在跨模态检索中，结合跨类代码检测的实际情况，提出了伴孪生网络架构。

通过不同的方式进行计算基本块嵌入，可以对不同语言的特性分别进行学习，避免丢失信息。

通过使用相同的方式对图结构进行拓扑计算，使得图计算过程中的模型参数能够被共享，这不仅减少了模型训练时所需要训练的参数，还减少了模型所需要学习的特征空间。

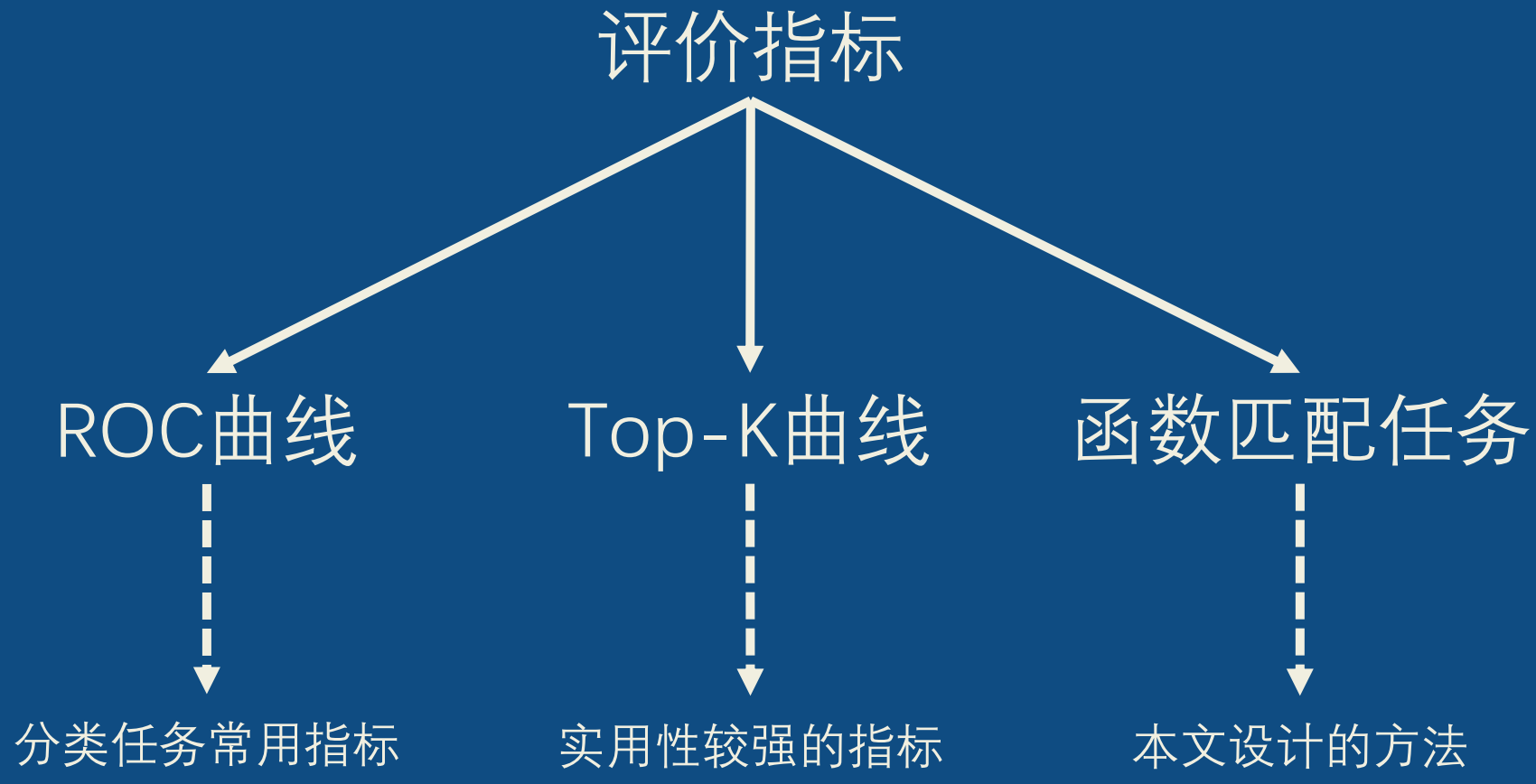
添加代码序列语义特征

在之前的代码相似性比较的工作中，大部分工作使用的是代码字面量作为节点的特征。由于缺乏语义特征，使得这些工作在代码功能性相似上很难更近一步。

GSN在代码字面量地基础上，引入了代码语义序列特征，并使用合适的结构进行学习，解决了函数特征缺乏代码功能性语义的问题。

華中科技大學
网络空间安全学院
School of Cyber Science and Engineering, HUST

实验结果与结论


评价指标
ROC曲线
Top-K曲线
函数匹配任务
分类任务常用指标
实用性较强的指标
本文设计的方法

ROC曲线缺陷

假设在一个实际测试中，正样例个数为N，负样例个数为M，在某一阈值下ROC曲线对应的点为(FPR, TPR)

$$precision = \frac{TP}{TP+FP} = \frac{N*TPR}{N*TPR+M*FPR} = \frac{1}{1+\frac{M}{N}*\frac{FPR}{TPR}}$$

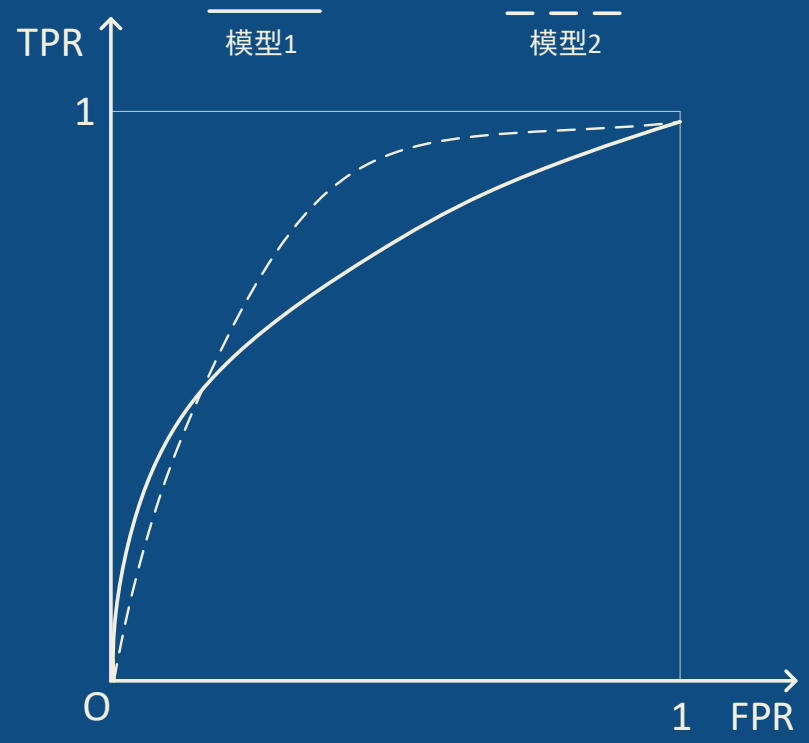
源函数集合X的大小为x，二进制函数集合Y的大小为y

$$precision = \frac{1}{1+\frac{xy-\min(x,y)}{\min(x,y)}*\frac{FPR}{TPR}} = \frac{1}{1+[\max(x,y)-1]*\frac{FPR}{TPR}}$$

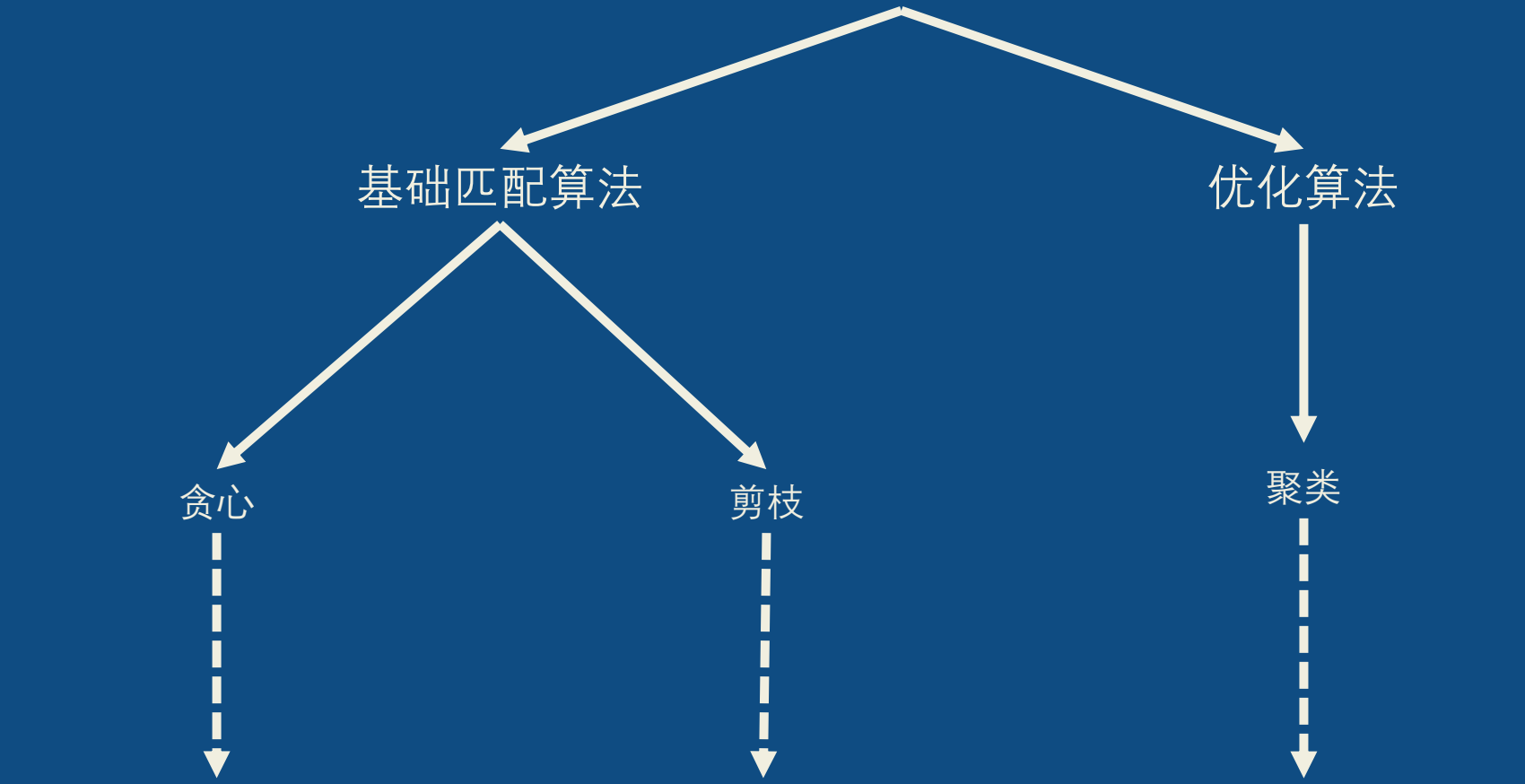
代数变换后

$$\frac{TPR}{FPR} = \frac{p}{1-p}*[\max(x,y)-1]$$

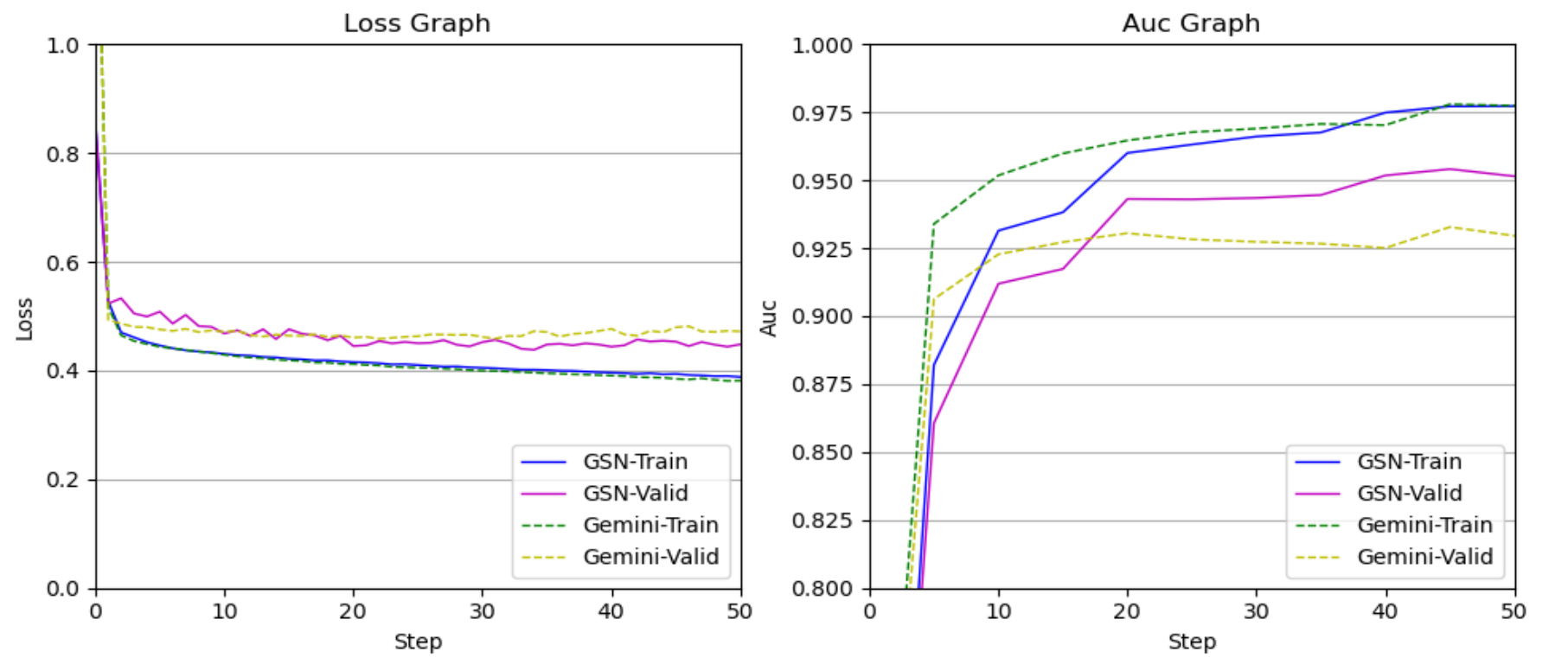
max(x,y)-1可以近似看作函数集规模。当精确率p达到50%以上时，TPR/FPR(点到原点的斜率)的值至少要达到L。这只占ROC曲线很小的一部分，ROC曲线的整体特性失去了作用



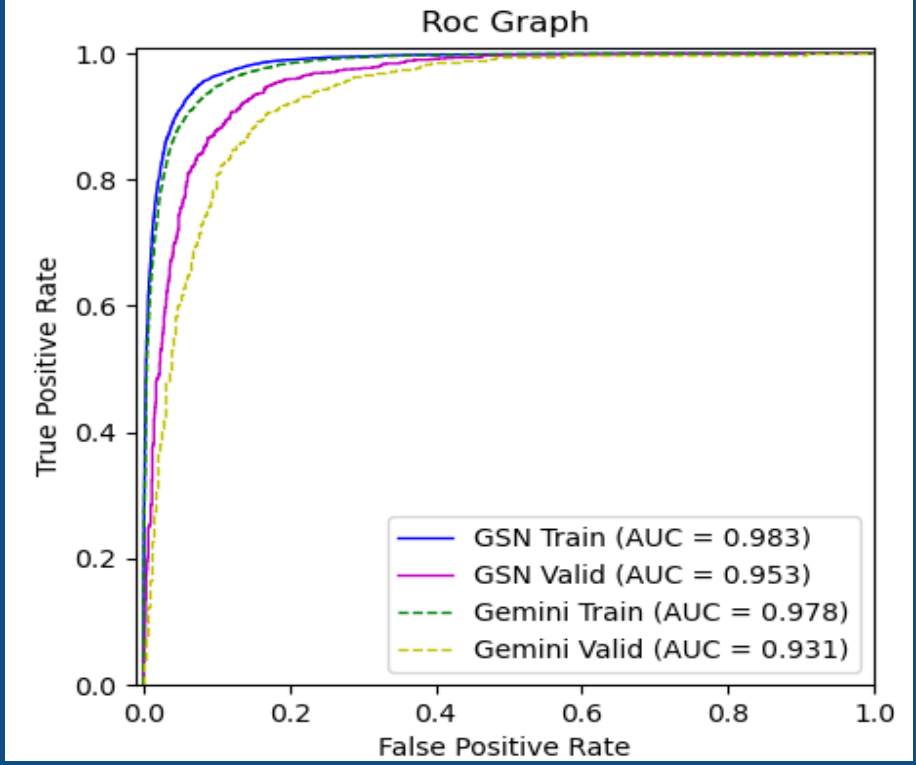
函数匹配任务

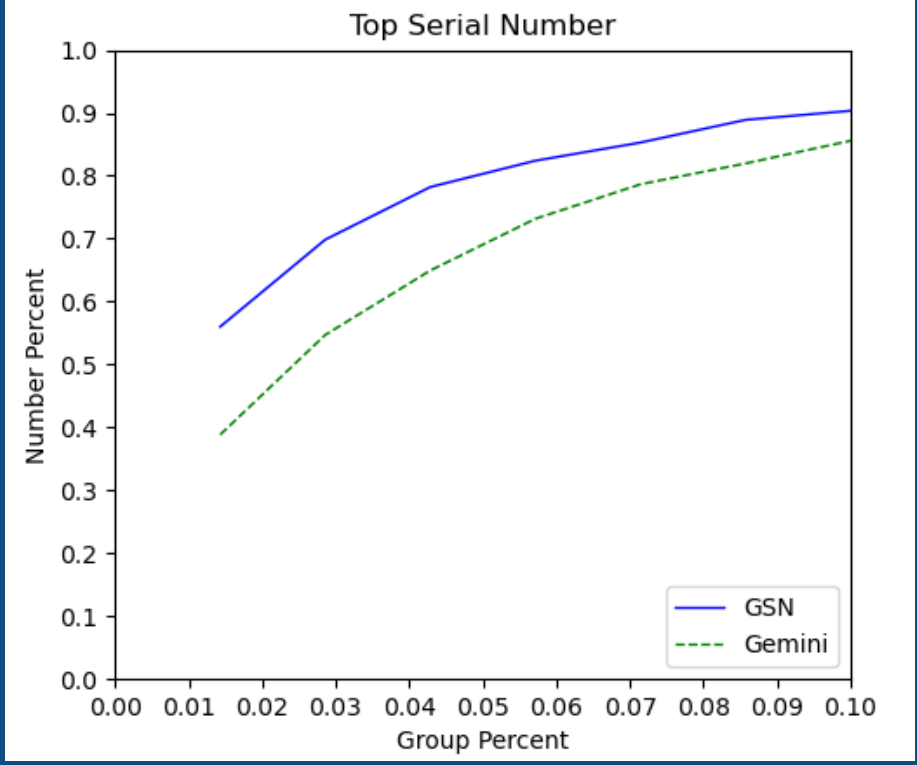

基础匹配算法
优化算法
贪心
剪枝
聚类
每次找出与次优匹配差值最大的一组匹配
相似度值小于某个值的匹配没有被选择的权力
先对匹配函数和被匹配函数进行聚类，然后优先在聚类内进行匹配

跨类代码相似性比较


Loss Graph
Loss
Step
GSN-Train
GSN-Valid
Gemini-Train
Gemini-Valid
Auc Graph
Auc
Step
GSN-Train
GSN-Valid
Gemini-Train
Gemini-Valid

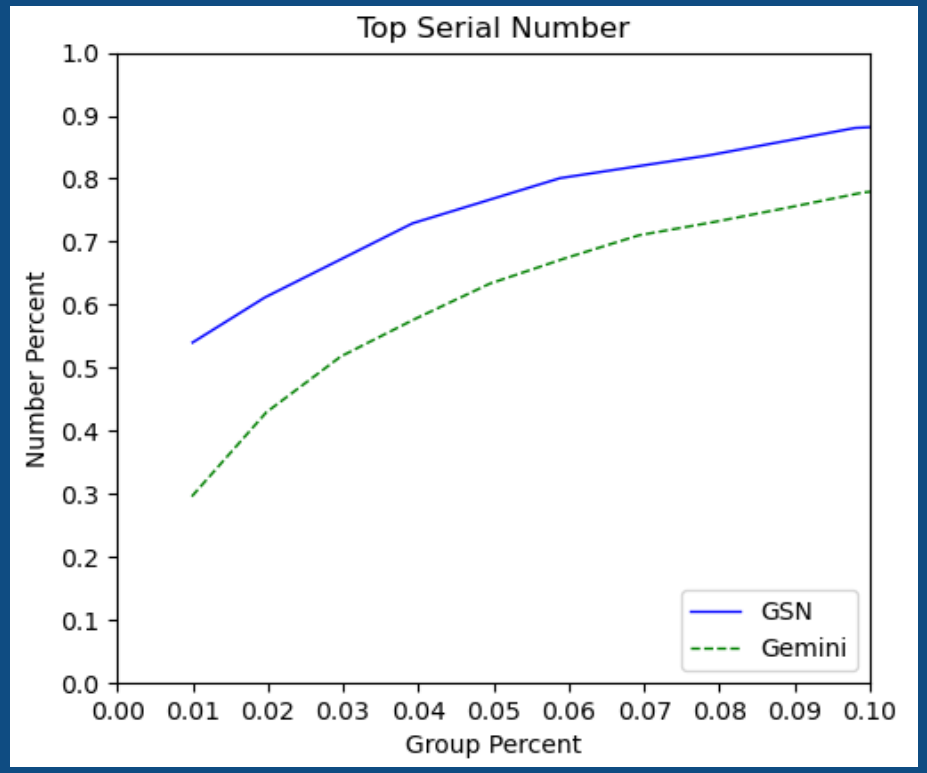
跨类代码相似性比较





GSN Top-1准确率	GSN 匹配准确率	Gemini Top-1准确率	Gemini 匹配准确率
43/100	60/100	25/100	35/100

二进制代码抗优化测试



模型改进点测试