CCF ChinaSoft
CCF 中国软件大会

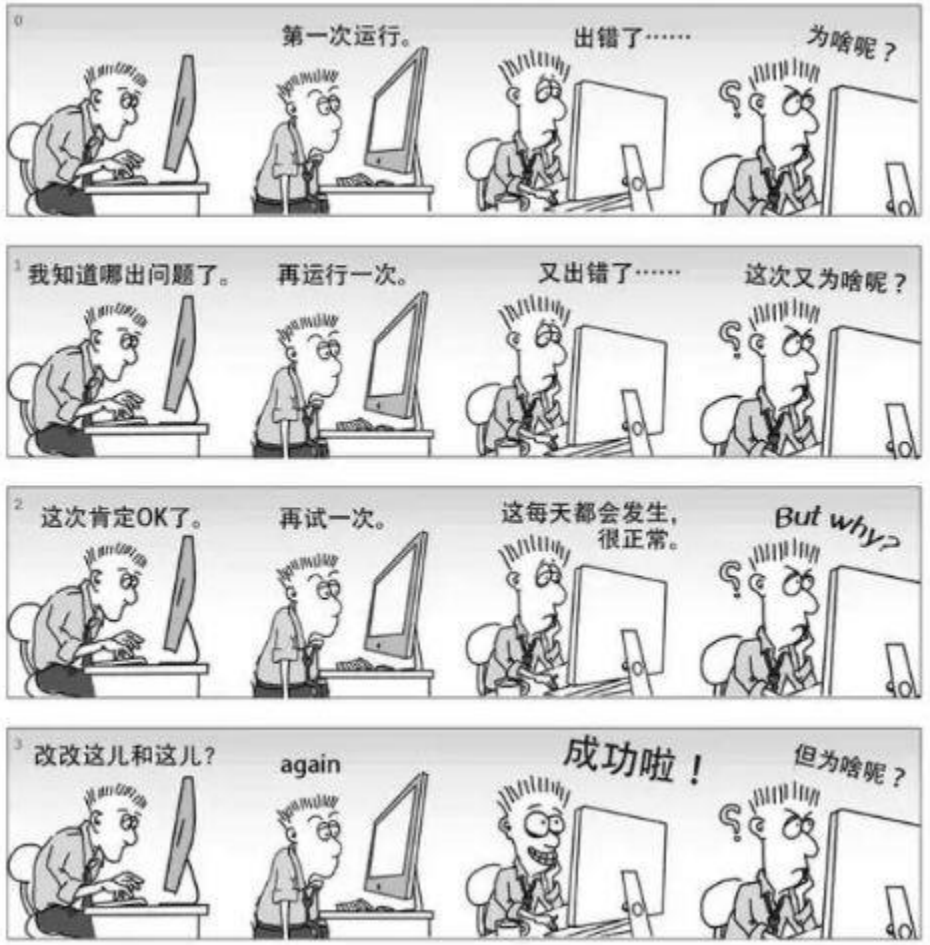
程序缺陷修复：历史与展望

报告人：熊英飞

报告日期：2022.11.25

程序员的人生

就是写Bug和修Bug交织在一起的悲歌



到底花了多少时间修Bug?

- 软件维护35.6%的时间是在修Bug[1]
- 软件维护成本通常认为占软件成本的90%
- 开发人员花在修复上的时间占全部开发时间一半左右[2]
- 开发团队可能没有足够资源修复所有缺陷[3]
- 软件在包含已知缺陷的情况下发布[4]

[1] B. P. Lientz, E. B. Swanson, and G. E. Tompkins, “Characteristics of application software maintenance,” Commun. ACM, vol. 21, no. 6, pp. 466–471, 1978
[2] Britton et al. Quantify the time and cost saved using reversible debuggers. Cambridge report, 2013
[3] J. Anvik, L. Hiew, and G. C. Murphy, “Coping with an open bug repository,” eXchange, 2005, pp. 35–39
[4] B. Liblit, A. Aiken, A. X. Zheng, and M. I. Jordan, “Bug isolation via remote program sampling,” in PLDI, 2003, pp. 141–154

缺陷自动修复的形式

- 输入：一个程序和其规约，并且程序不满足规约
- 输出：一个补丁，可以使程序满足规约

研究和实践中考虑最广泛的规约——
软件项目中的测试代码

缺陷自动修复的作用

- Yida Tao, Jindae Kim, Sunghun Kim, Chang Xu: Automatically generated patches as debugging aids: a human study. SIGSOFT FSE 2014: 64-74
  - 当程序员有正确的补丁做辅助的时候，修复正确率大幅提高，修复时间小幅减少
- Jingjing Liang, Ruyi Ji, Jiajun Jiang, Shurui Zhou, Yiling Lou, Yingfei Xiong, Gang Huang. Interactive Patch Filtering as Debugging Aid. ICSME'21: 37th International Conference on Software Maintenance and Evolution, September 2021.
  - 当有合适工具辅助的时候，即使生成错误补丁，仍然能大幅提升修复成功率（62.5%）和显著缩短修复时间（25.3%）

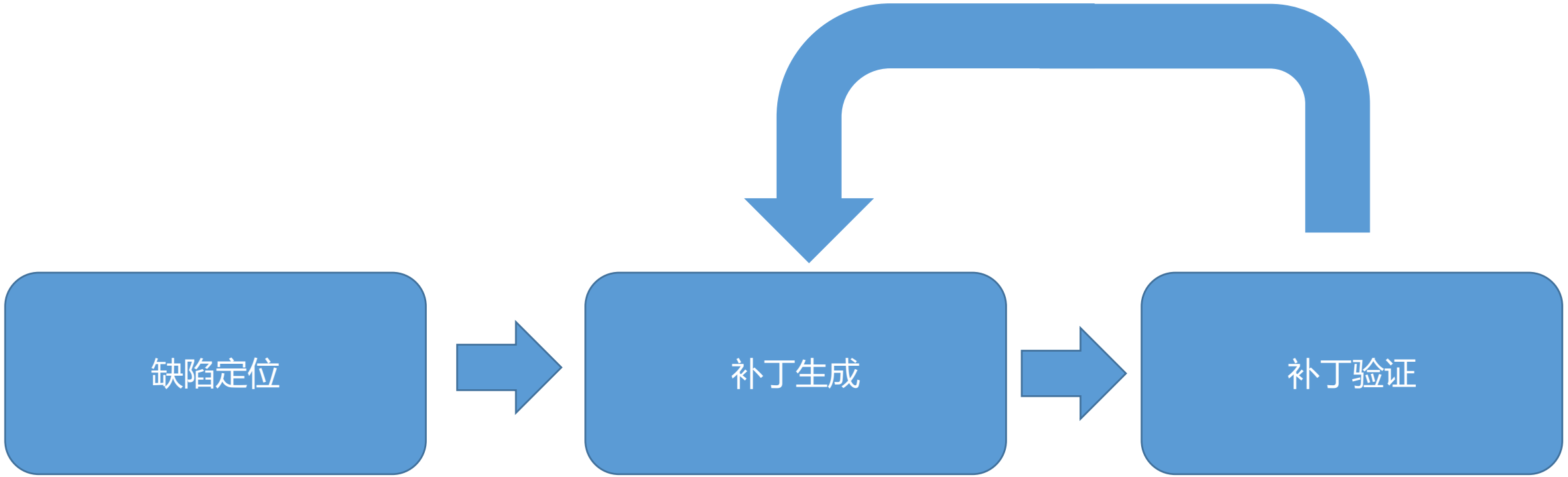
假设一个公司有5万程序员，年均人力成本支出为100万/人，修复技术对10%的缺陷生成补丁：
每年可节约人力成本：5万人*100万/年*35%*10%*25. 3%=4. 43亿元/年

工业界应用

- 阿里巴巴： Precfix
- Meta： SapFix、 Getafix
- 彭博社： Fixie
- 富士通： Elixir、 Hercules、 Phoenix
- 微软： LaMirage
- Google: TriCoder

- 其他没有发表论文的公司：华为、中兴、腾讯、360……

典型缺陷修复工具的流程



确定出错代码的位置

- 通常采用基于频谱的错误定位

生成对出错代码的修改

- 主要研究内容

检查修改是否正确

- 通常通过运行测试

缺陷修复的发展历程

- 2009年之前
- 2009-2014
- 2015-2020
- 2021-现在

史前时期（2009年之前）

- 修复一些特定类型的缺陷
  - 演化缺陷

- 修复一些特定类型的程序或软件制品
  - 布尔程序
  - 软件模型

探索时期（2009-2014）——成就

- 社会主义探索时期：建国-1966年
  - 取得辉煌的成就，也走了一些弯路
- 2009/2012年两篇GenProg论文
  - 大型软件上105个缺陷修复了55个
- 一系列代表性方法
  - 基于搜索的缺陷修复：GenProg, RSRepair
  - 基于模板的缺陷修复：PAR
  - 基于约束求解的缺陷修复：SemFix, Nopol, Angelix

探索时期（2009-2014）——中文社区贡献

RSRepair

- 国防科技大学毛晓光老师团队
- 首次发现GenProg中的遗传算法用处不大
- 替换成随机搜索效果反而更好

Nopol

- 武汉大学玄跻峰老师博后期间工作
- 首次成功把基于约束求解的方法用于实际项目缺陷

Contract-based Program Repair

- 香港理工大学裴玉老师博士期间工作
- 基于抽象状态分析产生补丁
- 全新体系，但技术难度大，曲高和寡，没有达到前三类工作的影响力

探索时期（2009-2014）——弯路

- 探索时期的工作以通过测试为目标
  - 各种不同方法都是为了生成能满足测试的补丁
- 但是，通过测试的补丁就是正确的吗?
- Zichao Qi, Fan Long, Sara Achour, Martin C. Rinard. An analysis of patch plausibility and correctness for generate-and-validate patch generation systems. ISSTA 2015
  - GenProg修复的55个缺陷中只有2个是正确的
  - 通过测试≠完整修复

发展时期（2015-2020）

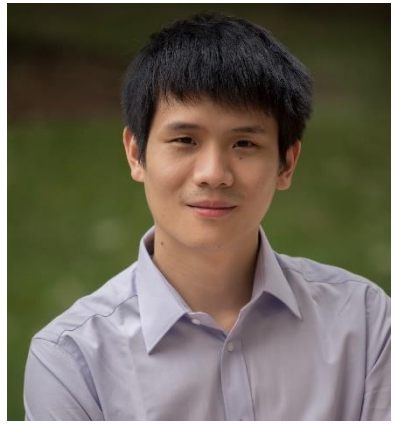
- 重新定义缺陷修复的指标
  - 正确率：产生的补丁中有多少是正确的
    - 实验中正确性一般通过和程序员写的补丁人工比较来判断
  - 召回率：在所有的缺陷中有多少是能够正确修复的
    - 仅计算人工判断正确的补丁
  - 修复效率：每个缺陷要花多少时间修复
- 正确率成为首要难题

“First open challenge.”
-- Claire Le Goues (CMU), ESEC/FSE, 2015

“Key discussion topic”
-- Dagstuhl Report 17022 “Automated Program Repair”

发展时期（2015-2020）——提升正确率

- 如何提升正确率?
- 2016年的系列探索
  - 反模式：禁止掉明显错误的补丁
  - Angelix：寻找修改最小的补丁
  - Prophet：基于历史补丁训练机器学习模型对补丁正确性打分
  - HDRepair：参考历史补丁变异出错代码
- 最好成绩：Prophet正确率38.5%



多伦多大学龙凡老师
博士期间提出Prophet

南方科技大学
陈馨慧老师
博士期间提出反模式

发展时期（2015-2020）——提升正确率

- 正确率关键提升
  - 2017年，北京大学熊英飞团队，ACS/L2S
    - 利用项目代码训练概率模型
    - 采用概率模型指导生成过程
    - 首次将修复正确率提升到73.9%
  - 2018年，香港科技大学张成志团队，CapGen
    - 基于上下文对变异算子和补丁进行精细排序
    - 将修复正确率进一步提升到84%

北京交通大学王博老师博士期间实现L2S

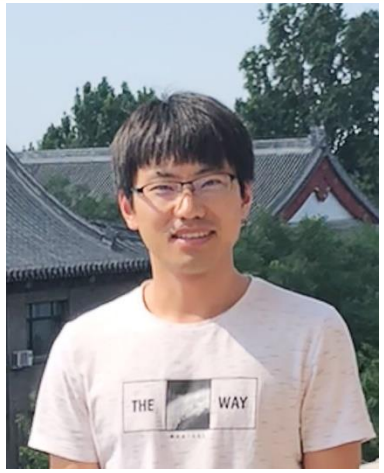
华中科技大学文明老师博士期间提出CapGen

发展时期（2015-2020）——提升正确率

- 进一步提升正确率的手段
  - 2018年，北京大学熊英飞团队，PatchSIM/TestSIM
    - 对于修复工具生成的补丁进一步分析过滤错误补丁
    - 可以采用更精细的分析
      - 补丁对测试执行过程的影响
    - 可将ACS正确率进一步提升至85%
    - 开辟了补丁正确性判断的子方向
      - 后续工作采用机器学习、深度学习等进一步判断补丁正确性

发展时期（2015-2020）——提升召回率

- 如何正确修复更多缺陷?
  - 寻找对大量缺陷有效的通用启发式规则
  - 2018年，北京大学熊英飞团队，SimFix
    - 通过参考相似代码进行修复
    - 在Defects4J1.2数据集上正确修复34个缺陷
  - 2019年，卢森堡大学Bissyandé团队，TBar
    - 在现有方法中找出最优模板集合
    - 在Defects4J1.2数据集上正确修复43个缺陷
  - 2019年，富士通Prasad团队，Hercules
    - 通过寻找代码克隆进行修复
    - 在Defects4J1.2数据集上正确修复46个缺陷

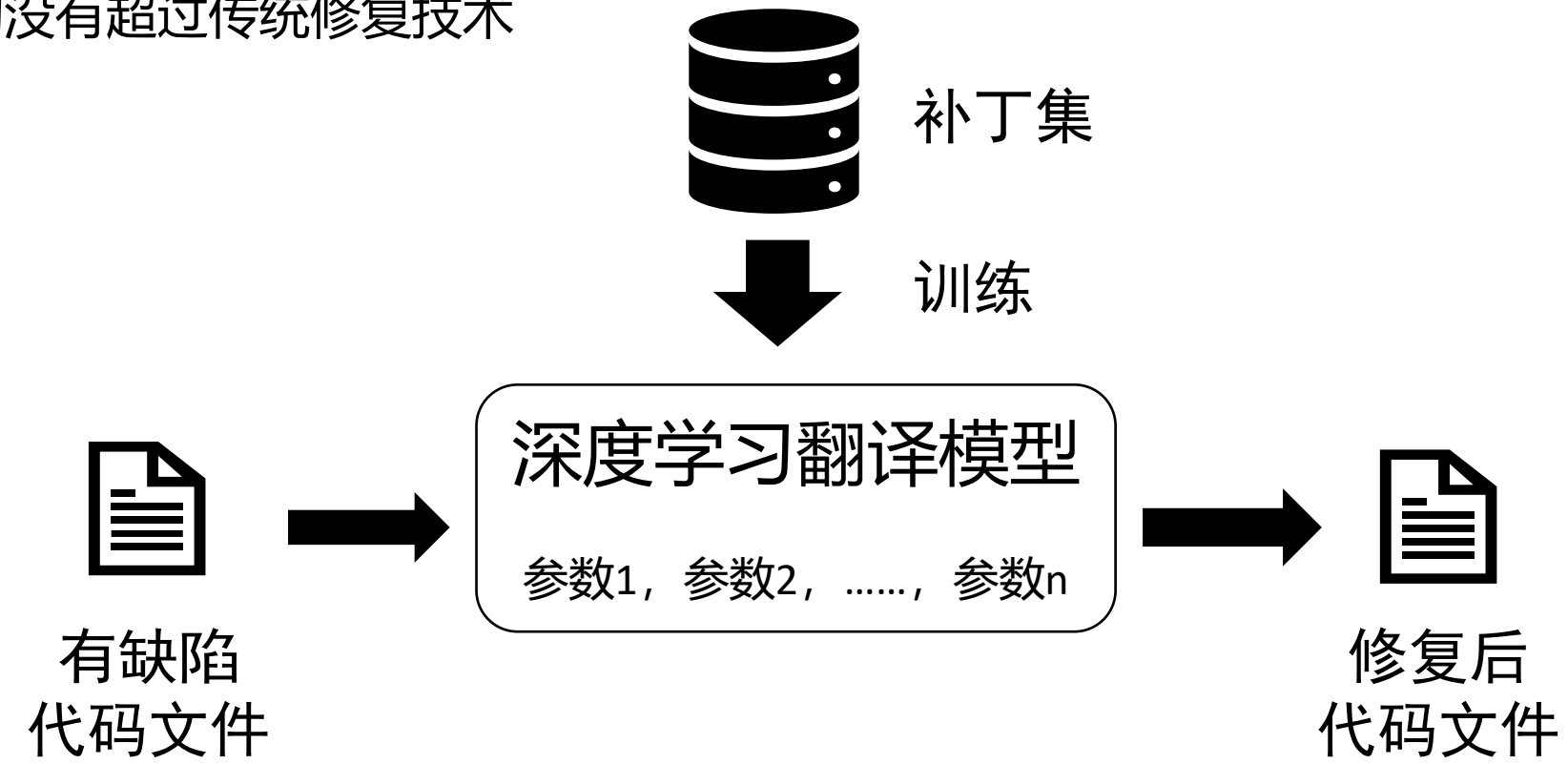


天津大学姜佳君老师
博士期间提出SimFix

华为技术专家刘逵老师
博士期间提出TBar

深度学习时期（2021-）

- 深度学习时代大量研究人员尝试用深度学习实现缺陷修复
  - 2017年开始大量工作：DeepFix, SequenceR, CODIT, CoCoNuT
  - 直接把修复问题看成深度学习翻译问题
  - 效果均没有超过传统修复技术



深度学习时期（2021-）

- 关键突破
  - 2021年，北京大学熊英飞团队，Recoder
    - 采用修改操作语法定义补丁空间
    - 神经网络引导语法展开
    - 在Defects4J1.2数据集上正确修复65个缺陷
  - 2022年，UIUC张令明团队，AlphaRepair
    - 采用大型预训练模型生成补丁
    - 在Defects4J1.2数据集上正确修复74个缺陷

北京大学博士生朱琪豪
Recoder提出者

美国伊利诺伊
大学香槟分校
张令明老师

提升修复效率

修复效率的瓶颈是对大量生成的补丁需要运行测试验证

国防科技大学毛晓光团队
只增量编译修改过的部分，节省大量编译时间[ICSM12]

北京大学熊英飞团队王博（现北交大讲师）提出
对大量补丁共享测试执行，状态相同的测试只执行一遍[ISSTA17]

美国UIUC张令明团队
通过直接在二进制代码上产生补丁避免编译开销[ASE19]
消除Java虚拟机启动开销[ICSE21]

提升错误定位效果

- 验证过程中补丁是否通过测试的信息可以用于指导错误定位
- 更准的错误定位可以生成更好的补丁
- 二者互相促进可形成更好的错误定位和修复方法

Retrospective Fault Localization

由UIUC张令明团队和
香港理工大学裴玉团队
同期并行提出

Unified Debugging

缺陷修复数据集

Defects4J -- version 2.0.0

build canceled

Defects4J is a collection of reproducible bugs and a supporting infrastructure with the goal of advancing software engineering research.

Contents of Defects4J

The projects

Defects4J contains 835 bugs from the following open-source projects:

Identifier	Project name	Number of bugs	Active bug ids	Deprecated bug ids (*)
Chart	jfreechart	26	1-26	None
Cli	commons-cli	39	1-5,7-40	6
Closure	closure-compiler	174	1-62,64-92,94-176	63,93
Codec	commons-codec	18	1-18	None
Collections	commons-collections	4	25-28	1-24
Compress	commons-compress	47	1-47	None
Csv	commons-csv	16	1-16	None
Gson	gson	18	1-18	None
JacksonCore	jackson-core	26	1-26	None

Defects4J：手动从项目历史中提取的补丁数据
美国华盛顿大学Just团队维护

General Introduction

This is a bug repository that keeps growing, called ***growingBugs***

Notably, each bug in ***growingBugs*** is composed of a buggy version, a fixed version, a ***concise patch*** (bug-fixing changes only), and one or more triggering test cases.

Contents of growingBugs

To date, growingBugs contains `1902` real-world bugs from open-source Java projects.

	Project ID	Project name	SubProject loc
1	Chart	jfreechart	
2	Cli	commons-cli	
3	Closure	closure-compiler	
4	Codec	commons-codec	
5	Collections	commons-collections	
6	Compress	commons-compress	
7	Csv	commons-csv	
8	Gson	gson	
9	JacksonCore	jackson-core	
10	JacksonDatabind	jackson-databind	

growingBugs：将手动提取方法自动化之后得到的大规模数据集

北京理工大学
刘辉教授团队

其他特定类型的缺陷修复

中科院软件所蔡彦老师
并行缺陷修复

北京大学郝丹老师
构建脚本修复

北京航空航天大学高祥老师
安全漏洞修复

总结与展望

- 过去15年
  - 缺陷修复社区从无到有，从实验室原型到工业界产品
  - 中国学术界从弱到强，从追随变为引领
  - 修复技术从没用变得可用，从很少能修变得时常能修
- 未来
  - 修复技术还需要从可用变得好用
  - 一方面进一步提升正确率、召回率和效率
    - 特别是降低大型神经网络模型方法的要求
  - 另一方面要打破原有方法的假设
    - 是否一定依赖测试?
    - 是否需要手动发现缺陷?
- 欢迎更多的人关注修复技术的发展!

感谢观看

Thank You