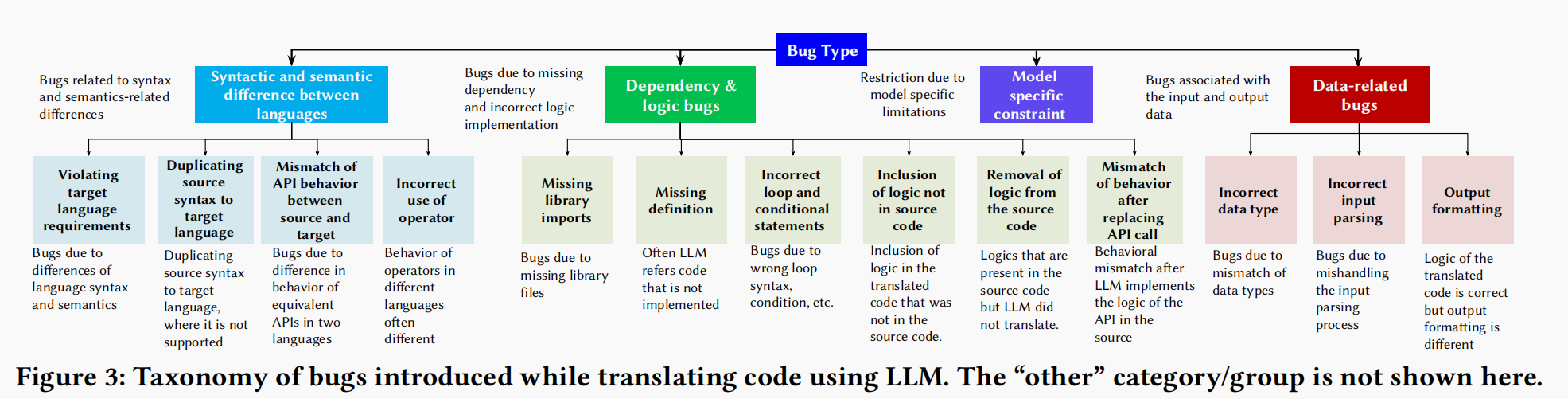
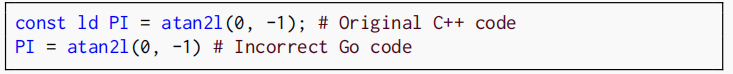
Lost in Translation: A Study of Bugs Introduced by Large Language Models while Translating Code



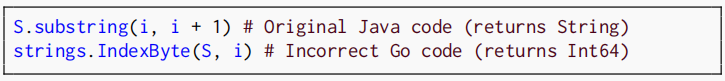
A.源语言与目标语言的句法/语义差异

A1.违反目标语言规范

A2.源语言语法残留（eg.目标语言不存在函数）



A3.API行为映射失配



A4.运算符误用

B.依赖与逻辑缺陷

B1.库导入缺失

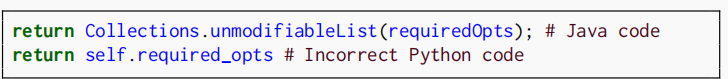
B2.定义(函数、结构体)缺失

B3.循环与条件语句误译

B4.逻辑冗余

B5.逻辑缺失

B6.API替换行为失配(A3)



C.数据相关缺陷

C1.错误数据类型

C2.输入解析错误

C3.输出格式问题

D.模型特定缺陷(与llm设计特性相关，eg.超出token限制)

E.其他缺陷(非llm引入)

cuda\_samples/0\_Introduction/

E → C3 – NVIDIA GPU架构获取函数，sycl不启用cuda后端无法得到相应信息以进行判断，从而无法输出架构信息。而翻译的源代码也没有尝试获取与判断。（在cuda\_samples中普遍存在）

E1--没有按照源代码的语义进行等价的翻译？？(eg.在simpleZeroCopy.cu程序中，源文件实现了无copy的数据传输，但翻译出的sycl程序没有遵循源码语义，直接进行memCopy)

文件名 源文件行数-迁移目标文件行数 编译是否正确

asyncAPI.cu 268-126 √

clock.cu 275-132 × A1\*3(错误传参、访问权限)A2\*4

simpleAssert.cu 238-125 × A1

vectorAdd.cu 146-97 √

simpleZeroCopy.cu 188-115 √ E1

cudaOpenMP.cu 109-122 √ E1

fp16ScalarProduct.cu 376-184 × A1\*2 A2\*6

(类似的代码导致同样错误出现两次)

simpleAttributes.cu 347-120 √ A2

(自定义获取设备信息导致运行时错误退出)

simpleAWBarrier.cu 394-154 × A1\*2 A2 B1 E

simplePrintf.cu 219-83 √

simpleHyperQ.cu 331-151 × A1\*3 A2\*2 B1

simpleOccupancy.cu 159-127 √ C1(数据溢出)

simpleCooperativeGroups.cu 90-91 × A2

编译正确率= 7/13 = 53.8%

freshman/

文件名 源文件行数-迁移目标文件行数 编译是否正确

hello\_world.cu 12-32 √ 输出正确

check\_dimension.cu 20-62 √ 输出正确

grid\_block.cu 24-30 √ 输出正确

sum\_arrays.cu 100-120 √ C3

sum\_arrays\_timer.cu 119-127 √ C3

thread\_index.cu 90-92 × A1 llm修复后正确

sum\_matrix.cu 164-191 × A1(cout<<range<2>)

device\_information.cu 88-60 × A2 C3

divergence.cu 129-114 √ C3

sum\_matrix2D.cu 138-140 × A1

reduceInteger.cu 239-159 × A1 C1(函数返回值声明出错)

reduceUnrolling.cu 462-250 × A1 C3

nested\_Hello\_World.cu 28-53 × A1

global\_variable.cu 19-43 √

pine\_memory.cu 97-116 √ C3

zero\_copy\_memory.cu 142-145 √

编译正确率= 9/16 = 56.3%